IJNAO

Iranian Journal of Numerical Analysis and Optimization

Volume 8 , Number 2

Summer 2018

Serial Number: 14

Ferdowsi University of Mashhad, Iran

In the Name of God

Iranian Journal of Numerical Analysis and Optimization (IJNAO)

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

Volume 8, Number 2, Summer 2018

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

Publisher: Faculty of Mathematical Sciences, Ferdowsi University of Mashhad

Published by: Ferdowsi University of Mashhad Press

Circulation: 50

Address: Iranian Journal of Numerical Analysis and Optimization

Faculty of Mathematical Sciences, Ferdowsi University of Mashhad

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

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

E-mail: ijnao@um.ac.ir

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

This journal is indexed by:

• Zentralblatt

• ISC

• SID

به اطلاع کلیه محققان، پژوهشگران، اساتید ارجمند، دانشجویان تحصیلات تکمیلی و نویسندگان محترم می رساند که نشریه ایرانی آنالیز عددی و بهینه سازی - IJNAO - طبق مجوز شماره ۳/۱۸/۵۴۸۹۱۳/ مورخه ۱۳۹۲/۱۰/۲۵ مدیر کل محترم سیاست گذاری و برنامه ریزی امور پژوهشی وزارت علوم، تحقیقات و فناوری، علمی – پژوهشی میباشد.
بر اساس نامه شماره ۹۲/۱۵۳۶/پ مورخه ۱۳۹۲/۱۱/۲۱ سرپرست محترم معاونت پژوهشی وفناوری پایگاه استنادی علوم جهان اسلام، نشریه ایرانی آنالیز عددی و بهینه سازی در پایگاه ISC نیز نمایه می شود.

Iranian Journal of Numerical Analysis and Optimization

Volume 8, Number 2, Summer 2018

Ferdowsi University of Mashhad - Iran

Iranian Journal of Numerical Analysis and Optimization

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

---

* Full Professor

** Associate Professor

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

Letter from the Editor in Chief

I would like to welcome you to the Iranian Journal of Numerical Analysis and Optimization (IJNAO). This journal is published biannually and supported by the Faculty of Mathematical Sciences at the Ferdowsi University of Mashhad. Faculty of Mathematical Sciences with three centers of excellence and three research centers is well-known in mathematical communities in Iran.

The main aim of the journal is to facilitate discussions and collaborations between specialists in applied mathematics, especially in the fields of numerical analysis and optimization, in the region and worldwide.

Our vision is that scholars from different applied mathematical research disciplines, pool their insight, knowledge and efforts by communicating via this international journal.

In order to assure high quality of the journal, each article is reviewed by subject-qualified referees.

Our expectations for IJNAO are as high as any well-known applied mathematical journal in the world. We trust that by publishing quality research and creative work, the possibility of more collaborations between researchers would be provided. We invite all applied mathematicians especially in the fields of numerical analysis and optimization to join us by submitting their original work to the Iranian Journal of Numerical Analysis and Optimization.

Ali R. Soheili

Contents

Iranian Journal of Numerical Analysis and Optimization
Vol. 8, No. 2, (2018), pp 1–23
DOI:10.22067/ijnao.v8i2.54962

Measurable functions approach for approximate solutions of Linear space-time-fractional diffusion problems

S. Soradi Zeid, A. V. Kamyad* and S. Effati

Abstract

In this paper, we study an extension of Riemann–Liouville fractional derivative for a class of Riemann integrable functions to Lebesgue measurable and integrable functions. Then we used this extension for the approximate solution of a particular fractional partial differential equation (FPDE) problems (linear space-time fractional order diffusion problems). To solve this problem, we reduce it approximately to a discrete optimization problem. Then, by using partition of measurable subsets of the domain of the original problem, we obtain some approximating solutions for it which are represented with acceptable accuracy. Indeed, by obtaining the suboptimal solutions of this optimization problem, we obtain the approximate solutions of the original problem. We show the efficiency of our approach by solving some numerical examples.

Keywords: Riemann–Liouville derivative; Fractional differential equation; Fractional partial differential equation; Lebesgue measurable and integrable function.

---

*Corresponding author
Received 9 April 2016; revised 14 December 2016; accepted 1 February 2017
S. Soradi Zeid
Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran. e-mail: s_ soradi@yahoo.com

A. V. Kamyad
Department of Applied Mathematics, School of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran. e-mail: avkamyad@yahoo.com

S. Effati
Department of Applied Mathematics, School of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran. e-mail: s-effati@um.ac.ir

1 Introduction

Fractional calculus is a generalization of classical calculus, which introduces derivatives and integrals of fractional order. Major reviews on the concepts and history of fractional calculus can be found in the book of [21].

Fractional differential equations (FDEs) have been found useful and applicable in science and engineering, such as physics, control theory, biology, finance, biomechanics, and electrochemical processes (see [2, 27] for more details). Most of these FDEs do not have analytic solutions; so there are some papers dealing with their numerical solutions, for example, predictor-corrector method [29], the Adomian decomposition [18], and the variational iteration method [19]. Podlubny in [22, 23] used the matrix expression to unify the formula, where the fractional derivatives are derived from a finite difference. Liu also did some interesting works on the numerical approach for the fractional differential equations in [17, 25] and a method based on collocation using spline functions given by [4].

On the other side, different models using FPDEs have been proposed in [3, 10], and there has been significant interest in developing numerical schemes for their solution. The great difficulty to obtain the numerical solutions of such problems is their solutions, which often are represented in terms of Mittag–Leffler functions, where these functions are in the form of series and their computation is not so easy. One method for solving the FPDE is pursued in the recent paper of [11]. They transformed this partial differential equation into a system of ordinary differential equations, which is then solved by using backward differentiation formulas. In another very recent paper, [26] proposed a finite difference method for the fractional diffusion-wave equation. [15] considered an implicit numerical scheme for fractional diffusion equation. The authors of [8] proposed a method for the solution of time fractional order partial differential equations by converting it into a nonlinear programming problem. [16] constructed an efficient spectral method for the numerical approximation of the space-time fractional diffusion equations, and [5] developed a finite element method for the time fractional Fokker–Planck equation. We refer the interested readers to see [12] for more information.

Based on the above review, we observe that the numerical methods for the FPDEs are abundant and when the function is only integrable are very limited. As far as we searched the related literature there were no reports on similar results as we are investigating in this paper. In this work we focus on a novel applicable approach with a better computational cost based on optimization problem. We consider a class of fractional convection-diffusion equations with measurable (or Lebesgue integrable) functions on the real line $\mathbb{R}$ and obtain a numerical approximation for the Riemann–Liouville derivative of Lebesgue measurable functions. Using the approximated functions whenever needed by a joint application of minimization the total error, we transform the original FPDE into a discrete optimization problem. By ob-

taining the optimal solutions of this problem, we obtain the approximate solution of the original problem. The results are more accurate and more useful than the ones introduced in [7] and should be very interesting for applications in sciences and engineering.

The discussion in the rest of paper will be as follows: in the next section, we introduce several definitions for different types of fractional derivatives of Lebesgue (or Riemann) integrable functions and the notation used in the numerical approximation for FPDE. The case with the FPDEs is displayed in section 3. Furthermore, in this section, we design an efficient approach to approximate the Riemann–Liouville fractional derivative and use it in our numerical approach. Finally, we will give some numerical examples in Section 4. Conclusions are included in the last section.

2 Preliminaries

First of all, let us introduce the properties of measurable functions.

2.1 Riemann–Liouville fractional extension to Lebesgue integral

Assume that (X, M, μ) is a fix measure space [6].

Definition 1. *If (X, M) is a measurable space and $A \subset X$, then the characteristic function χ_A of A is defined by:*

$$\chi_A(t) = \begin{cases} 1, & t \in A, \\ 0, & t \notin A. \end{cases} \tag{1}$$

It is easily checked that χ_A is measurable if and only if $A \in M$. A simple function on X is a finite linear combination, with complex coefficients, of characteristic functions of sets in M. Equivalently, $u : X \to \mathbb{C}$ is simple function if f is measurable and the range f is a finite subset of $\mathbb{C}$. Indeed, we have

$$u = \sum_{i=1}^{n} \gamma_i \chi_{A_i},$$

where $A_i = u^{-1}(\gamma_i)$, $1 \leq i \leq n$, and range(u)=$\{\gamma_1, \gamma_2, \ldots, \gamma_n\}$.

Theorem 1. *If $u : X \to [0, \infty)$ is measurable, there is a monotone increasing sequence $\{u_n\}$ of simple functions such that $0 \leq u_1 \leq u_2 \leq \cdots \leq u$, $u_n \to u$ pointwise, and $u_n \to u$ uniformly on any set on which u is bounded.*

Proof. See p. 47 of [6]. □

Theorem 2. *Let $u_n : X \to [0, \infty)$ be an increasing sequence of measurable functions on X and for a.e $x \in X$, $u(x) = \lim_{n\to\infty} u_n(x)$. Then:*

$$\int_X u = \lim_{n\to\infty} \int_X u_n.$$

Proof. See p. 76 of [24]. □

Theorem 3. *If $u_n : X \to [0, \infty)$ is a finite or infinite sequence of measurable functions on X and $u(x) = \sum_n u_n(x)$ for a.e $x \in X$, then $\int_X u = \sum_n \int_X u_n$.*

Proof. See p. 51 of [6]. □

In the special case when the μ is Lebesgue measure on $\mathbb{R}$, the integral, which we have developed, is called the Lebesgue integral. At this point it is appropriate to study the relation between the Lebesgue integrals and the Riemann integrals on $\mathbb{R}$.

Theorem 4. *Let u be a bounded real-valued function on $[a, b]$.*

(a) If u is Riemann integrable, then u is Lebesgue measurable and $\int_a^b u = \int_{[a,b]} u$.

(b) u is Riemann integrable if and only if $\{x \in [a, b] : u \text{ is discontinuous at } x\}$ has Lebesgue measure zero.

Proof. For more details of this theorem, see p. 57 of [6]. □

We shall generally use the notation $\int_a^b u(t)dt$ for Lebesgue integrals. Now, to compute the Lebesgue integral of u, we define the following partition on $[0, 1]$.

Definition 2. *We define the regular partition P_n on $[0, 1]$ by:*

$$P_n = \Big\{0 = \delta_0, \delta_1, \cdots, \delta_n = 1\Big\},$$

where $\delta_0 < \delta_1 < \cdots < \delta_n$. Here, according to the regularization of the partitions for each $i = 0, 1, \ldots, n$, we have $\delta_i = \dfrac{i}{n}$ and the partition norm is defined as:

$$\|P_n\|_\infty = \max_i \Big\{|\delta_i - \delta_{i-1}|\Big\} = \frac{1}{n}.$$

In the following, we introduce the definitions of fractional integral and derivatives of Lebesgue measurable and integrable functions that can be seen as a generalization of the classical derivative.

2.2 Fractional calculus

There are several different ways to define the fractional derivatives, and the most commonly used fractional derivatives are the Grunwald–Letnikov derivative, the Riemann Liouville derivative, and the Caputo derivative. For the definitions of fractional derivatives and some of their applications, see [14].

Let $[a,b] \in \mathbb{R}$. We will denote the space of all measurable and Lebesgue integrable real functions defined on $[a,b]$ by $L_1[a,b]$; that is,

$$L_1[a,b] = \Big\{u : \ \|u\|_{L_1[a,b]} = \int_a^b |u(t)|dt < \infty\Big\},$$

and the space of all measurable and essential bounded real functions defined on $[a,b]$ is denoted by $L^\infty[a,b]$.

Definition 3. *[13]. The fractional integral operator of order $\alpha > 0$ of a function $u(.) \in L_1([a,b],\mathbb{R}^n)$, is defined as:*

$$ {}_aI_t^\alpha u(t) = \frac{1}{\Gamma(\alpha)} \int_a^t (t-\tau)^{\alpha-1} u(\tau) d\tau, \tag{2}$$

where Γ is the classical gamma function and the integral of (2) exists and is in the sense of Riemann integral.

We may claim that the fractional integral (2) is a linear operator on the space of Lebesgue measurable functions; that is, if $f, g \in L_1[a,b]$ and C is a constant, then:

$$ {}_aI_t^\alpha (Cf+g) = C\ {}_aI_t^\alpha (f) + \ {}_aI_t^\alpha (g).$$

Proposition 1. *The fractional integral operator ${}_aI_t^\alpha$, $0 < \alpha < 1$, is bounded or equivalently continuous in $L_1[a,b]$:*

$$\| \ {}_aI_t^\alpha (u)\|_{L_1[a,b]} \le P_\alpha \|u\|_{L_1[a,b]}, \qquad P_\alpha = \frac{(b-a)^\alpha}{\Gamma(\alpha+1)}.$$

Proof. See its proof in [13]. □

Definition 4. *Let $u(.) \in L_1[a,b]$. The Riemann–Liouville fractional derivative (RLFD) of order $\alpha > 0$ is defined as:*

$$ {}_aD_t^\alpha u(t) = \frac{d^m}{dt^m}\ {}_aI_t^{m-\alpha} u(t) = \frac{1}{\Gamma(m-\alpha)} \frac{d^m}{dt^m} \int_a^t (t-\tau)^{m-\alpha-1} u(\tau) d\tau, \tag{3}$$

where $m-1 < \alpha \le m$ and $m \in \mathbb{N}$.

The above fractional derivative is equal to the fractional derivative of Riemann-integrable functions just on the compact subsets of $\mathbb{R}$. On the other hand, there are some functions that are only measurable and not Riemann

integrable but its derivations compute from (3). In this case, consider the following function:

$$f(t) = \begin{cases} 1, & t \in \mathbb{Q} \cap [0,1], \\ -1, & t \in \mathbb{Q}^c \cap [0,1]. \end{cases} \tag{4}$$

We know that $f(.)$ is not Riemann integrable function but its Lebesgue integrable. Now, we want to compute the fractional derivative of $f(.)$, for $0 < \alpha < 1$, as follows:

$$\begin{aligned} {}_0D_t^{\alpha} f(t) &= \frac{1}{\Gamma(1-\alpha)} \frac{d}{dt} \int_0^t (t-\tau)^{-\alpha} f(\tau) d\tau \\ &= \frac{1}{\Gamma(1-\alpha)} \frac{d}{dt} \Big\{ \int_{\mathbb{Q}\cap[0,t]} (t-\tau)^{-\alpha} d\tau - \int_{\mathbb{Q}^c\cap[0,t]} (t-\tau)^{-\alpha} d\tau \Big\}. \end{aligned}$$

As, $m([0,t]) = m([0,t] \cap \mathbb{Q}] + m([0,t] \cap \mathbb{Q}^c) = t$, we will have

$${}_0D_t^{\alpha} f(t) = \frac{-1}{\Gamma(1-\alpha)} \frac{d}{dt} \int_{\mathbb{Q}^c\cap[0,t]} (t-\tau)^{-\alpha} d\tau.$$

According to Theorem 4, the above statement is equivalent to Riemann integral and the result of this computation has been shown in Figure 1 for $\alpha = 0.5, 0.8, 0.9, 0.95$.

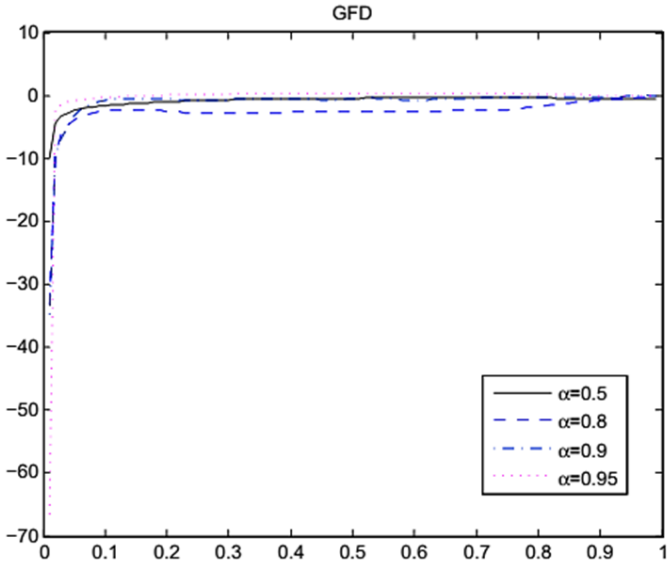


Figure 1: Fractional derivative of function (4) with different values of α

The following theorem help us to apply a fractional integral over a fractional derivative.

Theorem 5. *Let $\alpha > 0$.*

(1) ${}_aD_t^n\ {}_aD_t^\alpha u(t) = {}_aD_t^{n+\alpha} u(t)$, $n \in \mathbb{N}$, *where $\mathbb{N}$ is the set of natural numbers.*

(2) The equality ${}_aD_t^\alpha\ {}_aI_t^\alpha u(t) = u(t)$ *holds for every* $u(.) \in L_1(0,1)$.

(3) For $u(.) \in L_1(0,1)$, $n = [\alpha] + 1$, *if* ${}_aI_t^{n-\alpha} u \in AC^{n-1}[0,1]$, *then*

$$ {}_aI_t^\alpha\ {}_aD_t^\alpha u(t) = u(t) - \sum_{k=0}^{n-1} \frac{(t-a)^{\alpha-k-1}}{\Gamma(\alpha-k)} \Big(\frac{d^{n-k-1}}{dt^{n-k-1}}\ {}_aI_t^{n-\alpha} u \Big)(a), $$

in which $AC[a,b]$ denotes the space of absolutely continuous real functions on $[a,b]$, and

$$ AC^n[a,b] = \Big\{ u : [a,b] \to \mathbb{R} : \frac{d^{n-1}u(t)}{dt^{n-1}} \in AC[a,b] \Big\}. $$

In particular, if $0 < \alpha \leq 1$ *and* ${}_aD_t^\alpha u(t) \in L_1[a,b]$, *then*

$$ {}_aI_t^\alpha\ {}_aD_t^\alpha u(t) = u(t) - \frac{(t-a)^{\alpha-1}}{\Gamma(\alpha)} \Big({}_aI_t^{1-\alpha} u(a) \Big). $$

Proof. See pp. 241–256 of [13]. □

Now, we introduce the definitions of Caputo fractional derivative as follows.

Definition 5. *[13]. Let $\alpha > 0$, $n = \lceil \alpha \rceil$, and $u \in AC^n[a,b]$. The Caputo derivative of order $\alpha > 0$ is defined as*

$$ {}_a^C D_t^\alpha u(t) = {}_aI_t^{n-\alpha} \frac{d^n}{dt^n} u(t) = \frac{1}{\Gamma(n-\alpha)} \int_a^t (t-\tau)^{n-\alpha-1} \frac{d^n u(\tau)}{d\tau^n} d\tau. $$

In the space of the functions belonging to $AC^n[a,b]$ the following relation between Riemann–Liouville and Caputo derivative holds [9].

Theorem 6. *For $u \in AC^n[a,b]$, $n = \lceil \alpha \rceil$, the Riemann–Liouville derivative of order α of u exists almost everywhere and it can be written as*

$$ {}_aD_t^\alpha u(t) = {}_a^C D_t^\alpha u(t) + \sum_{k=0}^{n-1} \frac{(x-a)^{k-\alpha}}{\Gamma(k-\alpha+1)} u^{(k)}(a). \tag{5} $$

In order to simplify our discussion, throughout this paper, we will consider the Riemann–Liouville definition, since most useful tools have been

established by using the Riemann–Liouville definition. It is worthwhile to note, by virtue of (5), that for the homogeneous condition considered here the Riemann–Liouville definition coincides with the Caputo version.

3 Numerical approximation of FPDEs

Consider the space-time fractional convection-diffusion equation in one dimension as follows:

$$\frac{\partial^\alpha u(x,t)}{\partial t^\alpha} - \frac{\partial^\beta u(x,t)}{\partial x^\beta} = f(x,t), \quad x,t \in [0,1], \alpha \in (0,1],\ \beta \in (1,2], \tag{6}$$

subject to the following initial and boundary conditions:

$$u(x,0) = g(x), \qquad x \in [0,1], \tag{7}$$

$$u(0,t) = u(1,t) = 0, \qquad t \in [0,1], \tag{8}$$

where g is given a smooth function and f is a measurable function on $[0,1]$. Here, $\frac{\partial^\alpha}{\partial t^\alpha}$ is defined as the RLFD of order α and for all $x \in [0,1]$ it is given by:

$$\frac{\partial^\alpha}{\partial t^\alpha} u(x,t) = \frac{1}{\Gamma(1-\alpha)} \frac{d}{dt} \int_0^t u(x,s)(t-s)^{-\alpha} ds \quad t \in [0,1], \tag{9}$$

and for all $t \in [0,1]$, $\frac{\partial^\beta}{\partial x^\beta}$ is the RLFD of order β that is given by:

$$\frac{\partial^\beta}{\partial x^\beta} u(x,t) = \frac{1}{\Gamma(2-\beta)} \frac{d^2}{dx^2} \int_0^x u(\xi,t)(x-\xi)^{1-\beta} d\xi \quad x \in [0,1]. \tag{10}$$

Now, we assume that the integrals in equations (9) and (10) are well-defined. Also, suppose that the function u satisfies the Carathéodory conditions with respect to $L_1[0,1]$, which means, a non-negative measurable function is positive on a set of positive measure or, equivalently, the following conditions hold:

(C_1) For each $x \in [0,1]$, the mapping $t \mapsto u(x,t)$ is Riemann integrable.

(C_2) For a.e $t \in [0,1]$, the mapping $x \mapsto u(x,t)$ is continuous on $[0,1]$.

(C_3) Let $u(x,t) : [0,1] \times [0,1] \to [0,\infty)$ be a bounded function a.e on $L_1[0,1]$.

A standard assumption of continuity of the inhomogeneous term can be replaced with that of satisfying the Carathéodory conditions with respect to $L_1[0,1]$.

With specific α and β in equation (6), when $\alpha = 1$ and $\beta = 2$, we have the following classical diffusion equation:

$$\frac{\partial u(x,t)}{\partial t} - \frac{\partial^2 u(x,t)}{\partial x^2} = f(x,t), \qquad x,t \in [0,1]. \tag{11}$$

In fact the time derivative and space derivative of integer order in (11) can be obtained by taking the limit $\alpha \to 1^-$ in (9) and $\beta \to 2^-$ in (10).

In the following, we present a numerical approximation for the Riemann–Liouville derivative of Lebesgue measurable functions such that these operators simplify to the classical RLFI, and Riemann–Liouville fractional derivatives (RLFD) and also the numerical method that gives an approximate solution to the fractional convection-diffusion equations with measurable (or Lebesgue integrable) functions on real line $\mathbb{R}$, studied in this paper.

In the analysis of the numerical approach that follows, we will assume that this space-time fractional convection diffusion equation has a unique and sufficiently smooth solution. To establish the numerical approximation, and in order to simplify the notation, we consider

$$\Upsilon_{\alpha}^{x,u}(t) = \int_0^t u(x,s)(t-s)^{-\alpha} ds, \qquad 0 < \alpha < 1. \tag{12}$$

$$\Lambda_{\beta}^{t,u}(x) = \int_0^x u(\xi,t)(x-\xi)^{1-\beta} d\xi, \qquad 1 < \beta < 2. \tag{13}$$

Let $t_n = n\Delta t$, $n = 0, 1, \ldots, K$, where $\Delta t := \frac{1}{K}$ is the regular partition of time interval and $x_j = j\Delta x$, $j = 0, 1, \ldots, N$ and $\Delta x := \frac{1}{N}$ is the regular partition of space interval and K and N are two positive natural numbers. To motivate the construction of the scheme, we now define two operators that will be focus on them. We use the following approximation at t_n for all $0 \le n \le K$,

$$\Big(\frac{d}{dt}\Upsilon_{\alpha}^{x,u}\Big)(t_n) \simeq \frac{1}{\Delta t}\Big[\Upsilon_{\alpha}^{x,u}(t_n) - \Upsilon_{\alpha}^{x,u}(t_{n-1})\Big], \tag{14}$$

and the following approximation at x_j, for all $0 \le j \le N$,

$$\Big(\frac{d^2}{dx^2}\Lambda_{\beta}^{t,u}\Big)(x_j) \simeq \frac{1}{\Delta x^2}\Big[\Lambda_{\beta}^{t,u}(x_{j-1}) - 2\Lambda_{\beta}^{t,u}(x_j) + \Lambda_{\beta}^{t,u}(x_{j+1})\Big]. \tag{15}$$

For each t_n and x_j, respectively, we need to calculate $\Upsilon_{\alpha}^{x,u}(t_n)$ and $\Lambda_{\beta}^{t,u}(x_j)$. So, first, we compute (12) by approximating $u(x,s)$ at the fixed constant x by simple functions. Therefore, for each $s \in [0,1]$, we define $U(x,s)$ as an approximation of $u(x,s)$ at the fixed constant x, such that

$$U(x,s) = \sum_{i=1}^{K} u(x,t_i)\chi_{A_i}(s), \tag{16}$$

in which A_i is a sequence of disjoint sets in $[0,1]$ and $t_i = \frac{i}{K}$, $i = 1, 2, \ldots, K$. Now, we prove the uniformly convergence theorem of our approach.

Theorem 7. *Let $u(x,t)$ satisfy conditions $(C_1)-(C_3)$, and let $U(x,t)$ be defined in (16) for the fixed constant x. If K tends to infinity, then $U(x,t)$ tends to $u(x,t)$ uniformly on $[0,1]$.*

Proof. The proof is obtained from Theorem 1. □

This theorem means where we choose K as a sufficiently big number, closeness $U(x,t)$ to $u(x,t)$, at the fixed constant x, is independent of $t \in [0,1]$. So, by using the approximation (16) for $u(x,s)$ and suppose that $s \in [0,t_n]$, we will have

$$I_\alpha(t_n) = \int_0^{t_n} \sum_{i=1}^{n} u(x,t_i)\chi_{A_i}(s)(t_n-s)^{-\alpha} ds,$$

that is an approximation to $\Upsilon_\alpha^{x,u}(t_n)$. So,

$$I_\alpha(t_n) = \sum_{i=1}^{n} u(x,t_i) \int_0^{t_n} (t_n-s)^{-\alpha}\chi_{A_i}(s) ds = \sum_{i=1}^{n} u(x,t_i) \int_{[0,t_n]\cap A_i} (t_n-s)^{-\alpha} ds.$$

We denote $[0,t_n] \cap A_i$ with $A_{i,n}$, and note that every bounded Riemann integrable function defined on a bounded interval is Lebesgue integrable and the two integrals are the same (Theorem 4). Then we have

$$I_\alpha(t_n) = \frac{1}{1-\alpha} \sum_{i=1}^{n} u(x,t_i)\Big[-(t_n-s)^{1-\alpha}\Big]\Big|_{A_{i,n}} = \frac{1}{1-\alpha} \sum_{i=1}^{n} u(x,t_i) W_{i,n}^{(\alpha)}, \tag{17}$$

where

$$W_{i,n}^{(\alpha)} = \Big[-(t_n-s)^{1-\alpha}\Big]\Big|_{A_{i,n}}$$

and $t_i = \frac{i}{K}$ for $i = 1,2,\ldots,n$.

Finally an approximation for (14) is given by $\frac{1}{\Delta t}[I_\alpha(t_n) - I_\alpha(t_{n-1})]$; that is,

$$\Big(\frac{d}{dt}\Upsilon_\alpha^{x,u}\Big)(t_n) \simeq \frac{1}{(1-\alpha)\Delta t} \sum_{i=1}^{n} u(x,t_i)\Big(W_{i,n}^{(\alpha)} - W_{i,n-1}^{(\alpha)}\Big).$$

Denote U_j^n as an approximation to the value $u(x_j,t_n)$ on grid point (x_j,t_n). Then we take the following approximation for time fractional derivative (9) appeared in problem (6) for the fixed instant $x_j = j\Delta x \in [0,1]$,

$$\delta_t^\alpha U_j^n = \begin{cases} \dfrac{(\Delta t)^{-\alpha}}{\Gamma(2-\alpha)} \sum_{i=1}^{n} u(x_j,t_i) q_{i,n}^{(\alpha)}, & 0<\alpha<1, \\ \dfrac{u(x_j,t_n)-u(x_j,t_{n-1})}{\Delta t}, & \alpha = 1, \end{cases} \tag{18}$$

in which $q_{i,n}^{(\alpha)} = W_{i,n}^{(\alpha)} - W_{i,n-1}^{(\alpha)}$, and then we have

$$q_{i,n}^{(\alpha)} = \begin{cases} (n-i-1)^{1-\alpha} - 2(n-i)^{1-\alpha} + (n-i+1)^{1-\alpha}, & i \leq n-1, \\ 1, & i = n. \end{cases} \tag{19}$$

Incidentally, we find that $|q_{i,n}^{(\alpha)}|$ are bounded for all $\alpha \in [0,1]$ and all $i \geq 1$, as proven in the following Lemma.

Lemma 1. *For all $\alpha \in (0,1]$ and all $i \geq 1$, it holds $|q_{i,n}^{(\alpha)}| \leq c$, where c is dependent on n and α.*

Proof. First, for $\alpha = 1$ a direct calculation shows that $q_{i,n}^{(\alpha)}$ is a constant, for all $i \geq 1$. Now we prove the lemma for $\alpha \in (0,1)$. It can be verified that

$$A_{i,n-1} \subset A_{i,n} \subseteq [0,t_n] \Longrightarrow m(A_{i,n-1}) < m(A_{i,n}) \leq t_n.$$

In addition, It is well known that if $A \subseteq B$ and f is a measurable function on A and B, then $\int_A f \leq \int_B f$, and by consequence, we have

$$0 \leq W_{i,n-1}^{(\alpha)} \leq W_{i,n}^{(\alpha)} \leq t_n^{2-\alpha}.$$

So the proof is completed. □

Lemma 2. *Let $U(t) = \sum_{i=1}^{K} u(t_i)\chi_{A_i}(s)$, as explained in (16), be a sum of non-negative monotone increasing of simple functions that approximates the function $u(t) \in L_1[0,1]$. Then $\delta_t^{\alpha}(U(t))$, which is achieved by (18), tends to $D_{0,t}^{\alpha}u(t)$ as K tends to infinity.*

Proof. Using Lemma 1, and taking into account the fact that the phrase $(t-s)^{-\alpha}$, $0 < \alpha < 1$, is a continuous and integrable function and $\chi_{A_i}(s)$ is a measurable function on $[0,t]$, and then according to the convergence theorems of measurable functions (see Theorem 2 and Theorem 7 of this paper), the proof is obvious. □

Similarly, we approximate $u(\xi,t)$ at the fixed instant t by simple functions

$$\begin{cases} U(\xi,t) = \sum_{i=1}^{N} u(x_i,t)\chi_{B_i}(\xi), \\ \cup_{i=1}^{N} B_i = [0,1], B_i \cap B_j = \emptyset; i \neq j,\ x_i = \frac{i}{N}, \end{cases} \tag{20}$$

where $B_{i,j} = [0,x_j] \cap B_i$, $B_i = [\frac{i-1}{N}, \frac{i}{N}]$, and $x_j = \frac{j}{N}$. Now, we can approximate the space fractional derivative (10) for the fixed instant $t_n = n\Delta t \in [0,1]$ to form

$$\delta_x^{\beta} U_j^n = \begin{cases} \dfrac{(\Delta x)^{-\beta}}{\Gamma(3-\beta)} \sum_{i=1}^{j+1} u(x_i,t_n) q_{i,j}^{(\beta)}, & 1 < \beta < 2, \\ \dfrac{u(x_{j-1},t_n) - 2u(x_j,t_n) + u(x_{j+1},t_n)}{\Delta x^2}, & \beta = 2, \end{cases} \tag{21}$$

in which $q_{i,j}^{(\beta)} = W_{i,j-1}^{(\beta)} - 2W_{i,j}^{(\beta)} + W_{i,j+1}^{(\beta)}$ and $W_{i,j}^{(\beta)} = \Big[-(x_j - s)^{2-\beta} \Big]\Big|_{B_{i,j}}$, and we have

$$q_{i,j}^{(\beta)} = \begin{cases} (j-i+2)^{2-\beta} - 3(j-i+1)^{2-\beta} + 3(j-i)^{2-\beta} - (j-i-1)^{2-\beta}, \ i \leq j-1, \\ -3 + 2^{2-\beta}, & i = j, \\ 1, & i = j+1. \end{cases} \tag{22}$$

Now, for obtaining the solution of space-time fractional convection-diffusion equation (6)–(8), we apply (18) and (21) to (6) and reach the following numerical method,

$$\delta_t^{\alpha} U_j^n - \delta_x^{\beta} U_j^n = f(x_j, t_n), \tag{23}$$

$$U_j^0 = g(x_j), \ U_0^n = U_N^n = 0, \quad j = 1, \ldots, N, \ n = 1, \ldots, K, \tag{24}$$

So, if $u(x,t)$ is a solution of system (6)-(8), equivalently it is a solution of system of equations (23)–(24). Hence, we mention the following main theorem of this section.

Theorem 8. *The truncation error for the numerical approximation* (23)–(24) *is of order* $O(\Delta x)^{2-\beta} + O(\Delta t)^{1-\alpha}$.

Proof. Let $u(x,t)$ be a solution to the fractional partial differential equation (6) in which satisfies the conditions (7) and (8). Then we have

$$\begin{aligned} \frac{\partial^{\alpha}}{\partial t^{\alpha}} u(x, t_n) &= \frac{1}{\Gamma(1-\alpha)} \frac{d}{dt} \Upsilon_{\alpha}^{x,u}(t_n) \\ &= \frac{1}{\Gamma(1-\alpha)} \frac{1}{\Delta t} \Big(\Upsilon_{\alpha}^{x,u}(t_n) - \Upsilon_{\alpha}^{x,u}(t_{n-1}) \Big) + \epsilon_1(t_n), \end{aligned} \tag{25}$$

in which $\epsilon_1(t_n) = O(\Delta t)$. Let us define the error $E(t)$, such that,

$$\Upsilon_{\alpha}^{x,u}(t_n) - \Upsilon_{\alpha}^{x,u}(t_{n-1}) = I_{\alpha}(t_n) - I_{\alpha}(t_{n-1}) + E_1(t). \tag{26}$$

So, we have

$$\frac{\partial^{\alpha}}{\partial t^{\alpha}} u(x, t_n) = \frac{1}{\Gamma(1-\alpha)} \frac{1}{\Delta t} \Big(I_{\alpha}(t_n) - I_{\alpha}(t_{n-1}) \Big) + \frac{1}{\Gamma(1-\alpha)} \frac{1}{\Delta t} E_1(t) + O(\Delta t). \tag{27}$$

Now, by using Lemma 1 we have $E_1(t) = O(\Delta t)^{2-\alpha}$. So it follows that:

$$\frac{\partial^{\alpha}}{\partial t^{\alpha}} u(x, t_n) = \frac{1}{\Gamma(1-\alpha)} \frac{1}{\Delta t} \Big(I_{\alpha}(t_n) - I_{\alpha}(t_{n-1}) \Big) + \frac{1}{\Gamma(1-\alpha)} O(\Delta t)^{1-\alpha} + O(\Delta t).$$

Therefore,

$$\Big(\frac{\partial^\alpha}{\partial t^\alpha}u\Big)(x_j,t_n)=\delta_t^\alpha U_j^n+O(\Delta t)^{1-\alpha}. \tag{28}$$

Similarly, for the fractional derivative of order β, we have

$$\Big(\frac{\partial^\beta}{\partial x^\beta}u\Big)(x_j,t_n)=\delta_x^\beta U_j^n+O(\Delta x)^{2-\beta}. \tag{29}$$

Finally, by using the numerical method (23)–(24), we will have

$$\|U_j^n-u_j^n\|_\infty=O(\Delta t)^{1-\alpha}+O(\Delta x)^{2-\beta}, \tag{30}$$

and the proof is completed. □

In the remainder of this section, it will be shown that equations (23)–(24) can be converted to an optimization problem. For this suggested approach we need the following theorem.

Proposition 2. *If $h:[a,b]\to[0,\infty)$ is a measurable function, the necessary and sufficient condition for $\int_a^b h=0$ is that $h\equiv 0$ a.e., on $[a,b]$.*

Proof. See p. 51 of [6]. □

So, according to Proposition 2, necessary and sufficient condition for $u(x,t)$ to be a solution of system (23)–(24) is that the optimal solution of the following problem is zero (see Theorem 1 of [1] for convergence):

$$\min\int_0^1\int_0^1\Big|\delta_t^\alpha U(x,t)-\delta_x^\beta U(x,t)-f(x,t)\Big|dtdx. \tag{31}$$

By trapezoidal method and using the ending point in any subinterval for approximating integrals and using approximations (18) and (21) for (31), we obtain the following discretized problem:

$$\min_u\frac{\Delta t\Delta x}{c}\sum_{j=1}^{N}\sum_{n=1}^{K}\Big|a\sum_{i=1}^{n}u(x_j,t_i)q_{i,n}^{(\alpha)}-b\sum_{i=1}^{j+1}u(x_i,t_n)q_{i,j}^{(\beta)}-cf(x_j,t_n)\Big|, \tag{32}$$

where $a=(\Delta x)^\beta\Gamma(3-\beta)$, $b=(\Delta t)^\alpha\Gamma(2-\alpha)$, $c=ab$, and $q_{i,n}^{(\alpha)}$, $q_{i,j}^{(\beta)}$ are as before. Now, we convert the problem (32) to a linear programming problem with the following change of variables (see Theorem 3 and Lemma 2 of [1] and [28] for more details):

$$\min_u \frac{\Delta t \Delta x}{c} \sum_{j=1}^{N} \sum_{n=1}^{K} \left(r_j^n + e_j^n \right) \tag{33}$$

$$s.t.\ a \sum_{i=1}^{n} u_j^i q_{i,n}^{(\alpha)} - b \sum_{i=1}^{j+1} u_i^n q_{i,j}^{(\beta)} - cf(x_j, t_n) = r_j^n - e_j^n$$

$$u_j^0 = g(x_j),\ u_0^n = u_N^n = 0,$$

$$r_j^n, e_j^n \geq 0, j = 1, \ldots, N,\ n = 1, \ldots, K.$$

Finally, by obtaining the solution of problem (33), we recognize the value of unknown variables.

4 Numerical examples

In this section, we give some numerical examples and apply the presented approximation for solving them. These test problems demonstrate the validity and efficiency of this techniques.

Example 1. We compute ${}_0D_t^{\alpha} x(t)$, with $\alpha = 0.5$, for $x(t) = t^4$ by approximation (18).

The exact formulas of the derivatives are derived from

$${}_0D_t^{0.5}(t^s) = \frac{\Gamma(s+1)}{\Gamma(s+1-0.5)} t^{s-0.5},$$

Figure 2 shows the results. Since the exact solution for this problem is known, we compute the approximation error by using the maximum norm for each K. Assume that $\bar{x}(t_n)$, $n = 1, \ldots, K$, are the approximated values on the discrete time horizon $t_1, \ldots, t_K$. Then the error is given by

$$E = \max_n \left(|x(t_n) - \bar{x}(t_n)| \right). \tag{34}$$

In the case of $\alpha = 0.5$, with various choices of K, the maximum absolute errors are computed by equation (34) and shown in Table 1.

Example 2. Consider the following fractional differential problem:

$${}_0D_t^{\alpha} x(t) = g(x(t), t), \tag{35}$$

where $0 < \alpha \leq 1$, $x(0) = 0$ and $g(x,t) = \dfrac{2}{\Gamma(3-\alpha)} t^{2-\alpha} - \dfrac{1}{\Gamma(2-\alpha)} t^{1-\alpha} - x(t) + t^2 - t$.

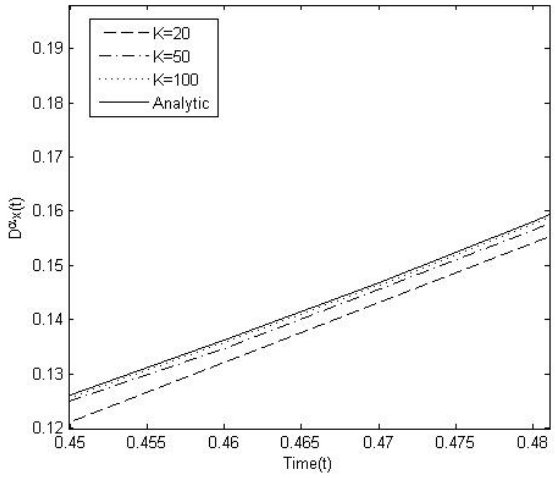


Figure 2: Analytic solution and numerical approximation (18), with various choices of K, for Example 1

Table 1: Maximum absolute error for Example 1

K	E_{appro}
5	2.65757×10^{-3}
10	6.47115×10^{-4}
20	4.09595×10^{-4}
40	9.77932×10^{-5}
50	2.27352×10^{-5}
100	5.04375×10^{-6}

The exact solution of this equation is $x(t) = t^2 - t$. We can rewrite the equation (35) as follows:

$$ {}_0D_t^{\alpha} x(t) - g(x(t), t) = 0. \tag{36} $$

Also, we have from equation (36) that

$$ \Big| {}_0D_t^{\alpha} x(t) - g(x(t), t) \Big| = 0. \tag{37} $$

By using again Proposition 2, we conclude that if $x(t)$ is a solution of equation (37) with the initial condition $x(0) = 0$, equivalently it is a solution of the following optimization problem:

$$ \min \int_0^1 \Big| {}_0D_t^{\alpha} x(t) - g(x(t), t) \Big| dt. \tag{38} $$

Then by discretization the integral as before an using equation (18) for approximate of ${}_0D_t^{\alpha}x(t)$, we simplify obtained problem (38) as follows:

$$\min_{x_i} \sum_{i=1}^{N}\sum_{k=1}^{K} \frac{(\Delta t)^{1-\alpha}}{\Gamma(2-\alpha)} \Big| \sum_{p=1}^{k} x_i(t_p) q_{p,k}^{(\alpha)} - (\Delta t)^{\alpha}\Big(\frac{2t_k^{2-\alpha}}{2-\alpha} - t_k^{1-\alpha} - \Gamma(2-\alpha)(x_i(t_k) - t_k^2 + t_k)\Big)\Big|, \tag{39}$$

where t_p and $q_{p,k}^{(\alpha)}$ are the same as before. Now, for solving the minimum problem (39), we can change it to a linear programming problem like (33). In Figures 3 and 4 we compare the exact solution with numerical approximation (39) for two different values of K, $N = 100$, and $\alpha = 0.5$.

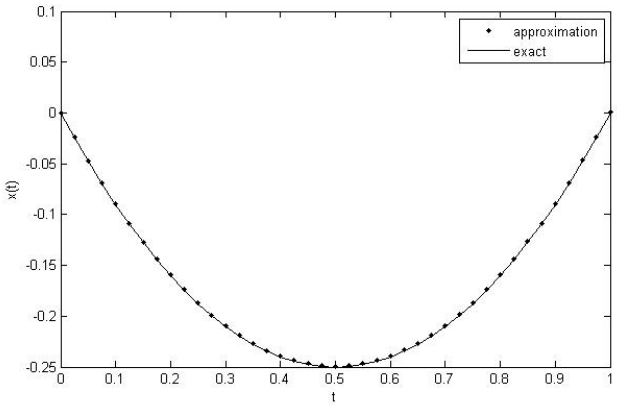


Figure 3: Analytic solution and numerical approximation (39) for Example 2 for $K = 40$

Table 2 shows the exact solution and the approximate solution for equation (35) by using problem (39) for $K = N = 100$ and $\alpha = 0.5, 0.99$. The results compare well with those obtained in [20]. From numerical results we can indicate that the solution of FDE approaches to the solution of integer order differential equation, whenever α approaches to its integer value.

Example 3. We consider the space-time fractional diffusion equation (6) with the following initial and boundary conditions:

$$u(x,0) = 0, \qquad 0 < x < 1,$$

$$u(0,t) = u(1,t) = 0, \qquad t \in [0,1],$$

where

$$f(x,t) = \frac{2}{\Gamma(3-\alpha)} t^{2-\alpha} sin(2\pi x) + 4\pi^2 t^2 sin(2\pi x).$$

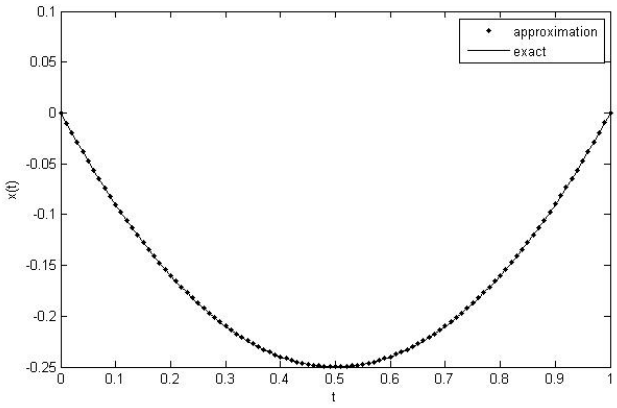


Figure 4: Analytic solution and numerical approximation (39) for Example 2 for $K = 100$

Table 2: Numerical values for Example 2 when $\alpha = 0.5, 0.99$

t	$x_{appro}(\alpha = 0.5)$	x_{exact}	$x_{appro}(\alpha = 0.99)$	x_{exact}
0.0	0.000000	0.000000	0.000000	0.000000
0.1	-0.089978	-0.090000	-0.089586	-0.090000
0.2	-0.159889	-0.160000	-0.159688	-0.160000
0.3	-0.209891	-0.210000	-0.209707	-0.210000
0.4	-0.239974	-0.240000	-0.239787	-0.240000
0.5	-0.249896	-0.250000	-0.249738	-0.250000
0.6	-0.239998	-0.240000	-0.239795	-0.240000
0.7	-0.199879	-0.210000	-0.209830	-0.210000
0.8	-0.160109	-0.160000	-0.159897	-0.160000
0.9	-0.096390	-0.090000	-0.100098	-0.090000

The exact solution of this problem when $\beta = 2$ is given by $u(x,t) = t^2 sin(2\pi x)$. The spatial and temporal meshes are taken uniform; that is, $\Delta t = \frac{1}{K}$, $\Delta x = \frac{1}{N}$. From the Table 3 and Figures 5, 6, and 7, we get that the approximate analytical solution is consistent with the exact solution when $\alpha \to 1$, $\beta \to 2$ and different values for K and N. Therefore, the validity of our numerical methods is confirmed.

Now, consider the vector $U(\Delta x) = (U_0, U_1, \cdots, U_N)$, where U_j is the approximate solution, for $x_j = j\Delta x$, $j = 0, 1, \ldots, N$ at a certain time t, and $u(\Delta x) = (u(x_0, t), \ldots, u(x_N, t))$, where u is the exact solution. The absolute error is defined as follows:

$$\|u(\Delta x) - U(\Delta x)\|_\infty = \max_{0 \le j \le N} |u(x_j, t) - U_j|. \tag{40}$$

Table 3: Maximum absolute error (40) for the numerical scheme of Example 3 at $t = 1$ for different values of α and $\beta = 2$ with various choices of Δt and Δx

	Δt	Δx	E_{appro}
	1/5	1/5	0.00847
$\alpha = 0.5$	1/10	1/10	0.00431
	1/15	1/15	5.32982×10^{-4}
	1/20	1/20	7.05965×10^{-5}
	1/5	1/5	0.00480
$\alpha = 0.9$	1/10	1/10	1.64153×10^{-4}
	1/15	1/15	7.19164×10^{-5}
	1/20	1/20	7.39623×10^{-6}
	1/5	1/5	0.00536
$\alpha = 0.95$	1/10	1/10	1.00857×10^{-4}
	1/15	1/15	3.38148×10^{-5}
	1/20	1/20	2.41435×10^{-6}
	1/5	1/5	0.00568
$\alpha = 0.98$	1/10	1/10	2.31643×10^{-4}
	1/15	1/15	4.62727×10^{-5}
	1/20	1/20	5.70822×10^{-6}
	1/5	1/5	0.00579
$\alpha = 0.99$	1/10	1/10	1.14488×10^{-4}
	1/15	1/15	9.19805×10^{-5}
	1/20	1/20	2.82570×10^{-6}

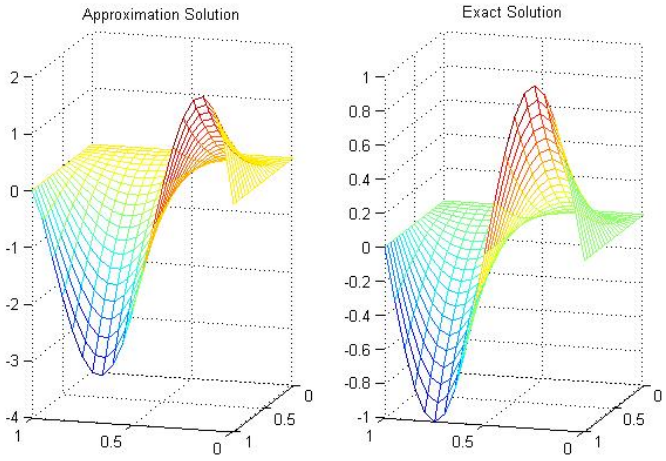


Figure 5: The evolution of $u(x, t)$ for anomalous diffusion coefficients $\alpha = 0.5$, $\beta = 2$ for Example 3

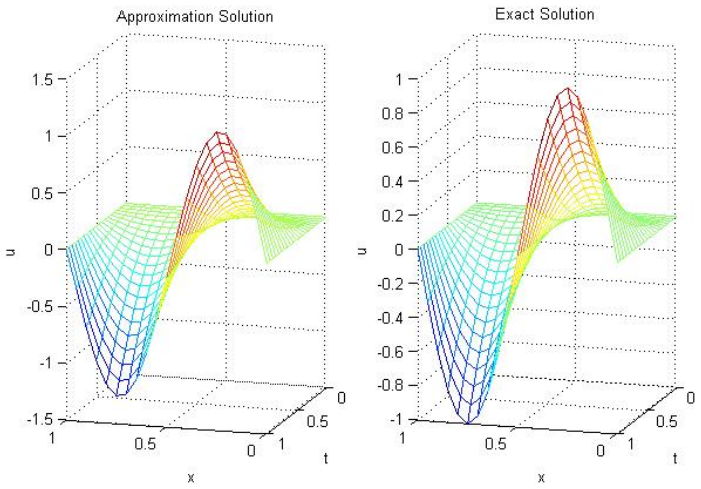


Figure 6: The evolution of $u(x,t)$ for anomalous diffusion coefficients $\alpha = 0.9$, $\beta = 2$ for Example 3

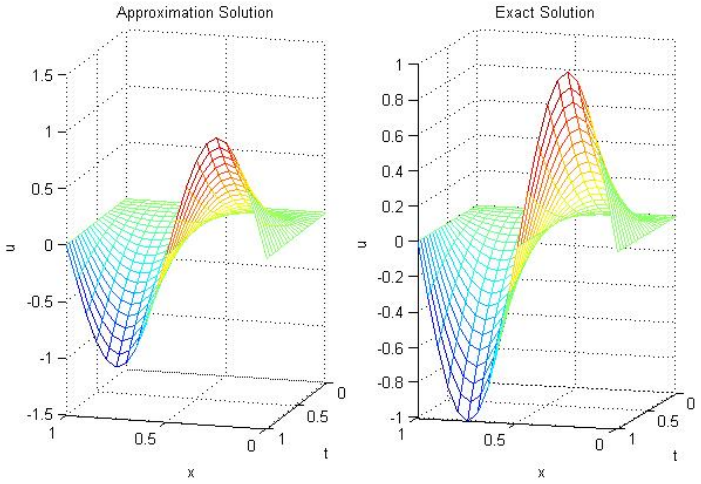


Figure 7: The evolution of $u(x,t)$ for anomalous diffusion coefficients $\alpha = 0.98$, $\beta = 2$ for Example 3

Table 4 presents a comparison between the absolute error of our method and the method presented by Neville et al. (2011) in [7] with different space and time steps. It should be noted that our numerical results, quickly converging to the exact solution with a lower division, are more accurate than with the results obtained in [16].

Table 4: Comparision of absolute error for the numerical scheme of Example 3 at $\alpha = 0.5$ and $\beta = 2$ with various choices of K, N

K	N	our method	K	N	estimated in [7]
5	5	0.000360	4	100	0.000926
10	10	0.000133	8	100	0.000337
15	15	5.12841×10^{-6}	16	100	0.000120
20	20	8.33727×10^{-6}	32	100	4.184925×10^{-5}

Example 4. we consider the following space-time fractional diffusion problem:

$$\frac{\partial^\alpha u(x,t)}{\partial t^\alpha} - \frac{\Gamma(2.8)x}{2}\frac{\partial^\beta u(x,t)}{\partial x^\beta} = q(x,t) - x^{0.8}\frac{\partial u}{\partial x}, \qquad (x,t) \in [0,1]\times[0,1] \tag{41}$$

with the initial condition:

$$u(x,0) = x^2(1-x), \qquad x \in [0,1], \tag{42}$$

and the boundary condition:

$$u(0,t) = u(1,t) = 0, \qquad t > 0, \tag{43}$$

where

$$q(x,t) = \frac{2x^2(1-x)^{\frac{1}{2}}}{\Gamma(2.2)} + 0.2x^{1.8}(1+t^2).$$

The exact solution of to this FPDE for $\alpha = 0.8$ and $\beta = 1.5$ is given by $u(x,t) = x^2(1-x)(1+t^2)$. Table 5 shows the absolute errors at time $t = 1$, between the analytical solution and the numerical solution obtained by applying the method discussed in this paper and the method presented in [30]. It should be noted that our results are very closely identical with other results.

Table 5: Comparision of absolute error for the numerical scheme of Example 4 for $\alpha = 0.8$ and $\beta = 1.5$ at $t = 1$ with various choices of K and N

K	N	our method	estimated in [30]
10	10	1.23×10^{-3}	1.26×10^{-3}
20	20	5.86×10^{-4}	6.74×10^{-4}
40	40	2.87×10^{-4}	3.48×10^{-4}
80	80	8.3×10^{-5}	8.6×10^{-5}

5 Conclusion

FDEs and FPDEs have caught considerable attentions due to their various applications. Since there is no systematic method to derive the exact solution of these equations, in this paper we partly solved a class of FPDE and obtained some approximating solution for it by provide an efficient approach based on measurable functions. For this purpose, at first, we present an approximation for the fractional derivatives of measurable functions. Then, by using this approximation and application of minimization the total error, we transform the original FPDE into a discrete optimization problem such that the optimal solutions of this problem is the approximate solution of the original problem. The suggested method represents a unifying approach for the solution of partial differential equations of fractional order.

From the numerical examples and the results that are compared with the exact solutions and with the other methods, it is shown that, as the number of discrete points, K and N, in proposed approach of this paper was increased, the solutions converged to the exact solutions. In addition, as the value of α approaches one and β approaches two, the numerical solutions for both the FDE and FPDE approach the analytical solutions for $\alpha = 1$ and $\beta = 2$. Since the proposed approximation of this paper is based on the minimization of total error, it is clear from the results that there is no difference between exact and approximate solution in point to point case.

References

1. Badakhshan, K. P. and Kamyad, A. V. *Numerical solution of nonlinear optimal control problems using nonlinear programming*, Appl. Math. Comput., 187 (2007), 1511–1519.

2. Bhrawy, A. H., Doha, E. H., Tenreiro Machado, J. A. and Ezz-Eldien, S. S. *An Efficient Numerical Scheme for Solving Multi-Dimensional Fractional Optimal Control Problems With a Quadratic Performance Index*, Asian J. Control 17(2015), no. 6, 2389–2402

3. Bhrawy, A. H., Zaky, M. A. and Machado, J. T. *Efficient Legendre spectral tau algorithm for solving the two-sided space-time Caputo fractional advection-dispersion equation*, J. Vib. Control, 22 (2016), no. 8, 2053–2068. .

4. Blank, L. *Numerical treatment of differential equations of fractional order*, Nonlinear World 4 (1997), no. 4, 473–491.

5. Deng, W. *Finite element method for the space and time fractional Fokker-Planck equation*, SIAM J. Numer. Anal. 47 (2008/09), no. 1, 204–226.

6. Folland, G. B. *Real analysis: modern techniques and their applications*, John Wiley & Sons, 2013.

7. Ford, N. J., Xiao, J. and Yan , Y. *A finite element method for time fractional partial differential equations*, Fract. Calc. Appl. Anal. 14 (2011), no. 3, 454–474.

8. Ghandehari, M. A. M. and Ranjbar, M. *A numerical method for solving a fractional partial differential equation through converting it into an NLP problem*, Comput. Math. Appl. 65 (2013), no. 7, 975–982.

9. Huang, G., Huang, Q. and Zhan, H. *Evidence of one-dimensional scale-dependent fractional advection-dispersion*, J. contam. hydro. 85 (2006), 53–71.

10. Irandoust-pakchin, S., Dehghan, M., Abdi-mazraeh, S. and Lakestani, M. *Numerical solution for a class of fractional convection-diffusion equations using the flatlet oblique multiwavelets*, J. Vib. Control 20(2014), no. 6, 913–924.

11. Jafari, H. and Tajadodi, H. *Numerical solutions of the fractional advection-dispersion equation*, Progr. Fract. Differ. Appl. 1, (2015), No. 1, 37–45.

12. Jia, J. and Wang, H. *Fast finite difference methods for space-fractional diffusion equations with fractional derivative boundary conditions*, J. Compu. Phys. 293 (2014), 359–369.

13. Kilbas, A. A. A., Srivastava, H. M. and Trujillo, J. J. *Theory and applications of fractional differential equations*, North-Holland Mathematics Studies, 204. Elsevier Science B.V., Amsterdam, 2006.

14. Kiryakova, V. S. *Generalized fractional calculus and applications*, Pitman Research Notes in Mathematics Series, 301. Longman Scientific & Technical, Harlow; copublished in the United States with John Wiley & Sons, Inc., New York, 1994.

15. Langlands, T. A. M. and Henry, B. I. *The accuracy and stability of an implicit solution method for the fractional diffusion equation*, J. Comput. Phys. 205 (2005), 719–736.

16. Lin, Y. and Xu, C. *Finite difference/spectral approximations for the time-fractional diffusion equation*, J. Comput. Phys. 225 (2007), 1533–1552.

17. Liu, F., Anh, V. and Turner, I. *Numerical solution of the space fractional Fokker-Planck equation*, J. Comput. Appl. Math. 166 (2004), 209–219.

18. Momani, S. and Odibat, Z. *Analytical solution of a time-fractional Navier-Stokes equation by Adomian decomposition method*, Appl. Math. Comput. 177 (2006), 488–494.

19. Odibat, Z. M. and Momani, S. *Application of variational iteration method to nonlinear differential equations of fractional order*, Int. J. Nonlin. Sci. Num. Simul. 7 (2006), 27–34.

20. Odibat, Z. and Momani, S. *An algorithm for the numerical solution of differential equations of fractional order*, J. Appl. Math. Inform, 26 (2008), 15–27.

21. Podlubny, I. *Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*, Academic press, 198 1998.

22. Podlubny, I. *Matrix approach to discrete fractional calculus*, Fract. Calc. Appl. Anal. 3 (2000), 359–386.

23. Podlubny, I., Chechkin, A., Skovranek, T., Chen, Y. and Jara, B. M. V. *Matrix approach to discrete fractional calculus II: Partial fractional differential equations*, J. Comput. Phys. 228 (2009), 3137–3153.

24. Rudin, W. *Real and complex analysis*, Tata McGraw-Hill Education, 1987.

25. Shen, S., Liu, F., Anh, V. and Turner, I. *The fundamental solution and numerical solution of the Riesz fractional advection-dispersion equation*, IMA J. Appl. Math. 73 (2008), 850–872.

26. Sun, Z. and Wu, X. *A fully discrete difference scheme for a diffusion-wave system*, Appl. Numer. Math. 56 (2006), 193–209.

27. Tricaud, C. and Chen, Y. Q. *An approximate method for numerically solving fractional order optimal control problems of general form*, Comput. Math. Appl. 59 (2010), 1644–1655.

28. Zeid, S. S., Kamyad, A. V., Effati, S., Rakhshan, S. A. and Hosseinpour, S. *Numerical solutions for solving a class of fractional optimal control problems via fixed-point approach*, SeMA J. 74 (2017), 585–603.

29. Zhao, L. and Deng, W. *Jacobian-predictor-corrector approach for fractional differential equations*, Adv. Comput. Math. 40 (2014), 137–165.

30. Zhang, Y. *Finite difference approximations for space-time fractional partial differential equation*, J. Numer. Math. 17 (2009), 319–326.

Iranian Journal of Numerical Analysis and Optimization
Vol. 8, No. 2, (2018), pp 25–37
DOI:10.22067/ijnao.v8i2.59531

A modified flux-wave formula for the solution of one-dimensional Euler equations with gravitational source term

H. Mahdizadeh*

Abstract

In this paper a novel Godunov-type finite volume technique is presented for the solution of one-dimensional Euler equations. The numerical scheme defined herein in is well-balanced and approximates the solution by propagating a set of jump discontinuities from each Riemann cell interface. The corresponding source terms are then treated within the flux-differencing of the finite volume computational cells. First, the capability of the numerical solver under gravitational source term is examined and the results are validated with reference solution and higher-order WENO scheme. Then, the well-balanced property of the scheme for the steady-state is tested and finally the proposed method is employed for the modeling small and large amplitude perturbation imposed to the polytropic atmosphere. It is found out that the defined well-balanced solver provides sensible prediction for all of the given test cases.

Keywords: Wave propagation algorithm, Flux wave formula, Riemann solver, Well-Balanced, Euler equations.

1. Introduction

The Euler equations with gravitational source terms have been extensively used in many scientific aspects such as aerospace, astrophysics, and shock tube problems. Prediction models should be able to capture sharp gradient shocks as well as rarefaction waves that appear within the solution in particular with existence of source terms. Generally, two different class of finite volume methods have been used to model Euler equations in recent

*Corresponding author
Received 14 October 2016; revised 31 January 2018; accepted 28 February 2018
H. Mahdizadeh
Department of Civil Engineering, University of Birjand, Iran.
e-mail: hossein.mahdizadeh@birjand.ac.ir

years mainly reviewed by LeVeque [6, 7]and [16]. The first method is upwind method, which basically uses the Godunov scheme. Despite the complexity of upwind schemes in particular in dealing with associated Jacobian matrix they provide very accurate results for the shock capturing problems [17]. The second approach is central schemes, which applies Lax–Friedrichs or Lax-Wendroff methods. However, they produce rather high diffusion, which eventually affects the methods stability unless a refined mesh is used [7].

Previously, significant attentions have been paid for the solution of the Euler equations with the gravitational source terms mostly based on the finite volume methods. LeVeque and Bale [8] have developed a quasi-state wave propagation algorithm for the solution of the Euler equations. In another work Botta et al. [2] defined a well-balanced finite-volume methods, which maintain certain class of steady states for the nearly hydrostatic flows. A well-balanced approach on the basis of the gas-kinetic scheme has been proposed by Luo et al. [10] for an isolated gravitational hydrodynamic system. Käppeli and Mishra [5] have introduced a second-order well-balanced finite volume scheme for the isentropic hydrostatic equilibrium. Chandrashekar and Klingenberg [3] implemented a well-balanced second-order Godunov-type finite volume method for the Euler equations with gravitation. More recently, Li and Xing designed a high-order well-balanced finite volume WENO (weighted essentially oscillatory) scheme for the Euler equations with gravitational field, which preserves both isothermal equilibrium and the polytropic hydrostatic balance state.

The main purpose of this work is to develop a version of Godunov-type wave propagation algorithm for the solution of one dimensional Euler equations. The proposed method extends the a modified flux-wave solution provided in [11, 12] for the shallow water equations (SWEs) to the one dimensional Euler equations. This approach is well-balanced and treats any source terms within the flux-differencing of the finite-volume computational cells. Additionally, the defined numerical scheme utilizes the advantage of combination both approximate and exact Riemann speeds, which enables the method to avoid non-negative pressure fields for the Euler equations. To the best of author's knowledge no development of the modified flux-wave approach defined in [11, 12] is used for the solution of the one dimensional Euler equations with gravitational source term. The rest of this paper is organized as follows: In the next section the mathematical equations for the one dimensional Euler equations consisting of gravitational source terms are provided. Then in the second section, the wave propagation algorithm with both first-order and high-resolution accurate terms is expressed. In the third section, the flux-wave formula comprising different choice of wave speeds for the one dimensional Euler equations is described. Fourth section states the validation of the introduced numerical method with the reference solutions and other numerical results available in literature. Finally, the paper ends with the summary of numerical results and conclusions of findings.

2. Governing equations

The one dimensional Euler equations including source terms can take the conservation law form as

$$\boldsymbol{U}_t + \boldsymbol{F}(\boldsymbol{U})_x = \boldsymbol{S},$$

$$\boldsymbol{U} = \begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix}, \ \boldsymbol{F}(\boldsymbol{U}) = \begin{bmatrix} \rho u \\ \rho u^2 + P \\ (E+P)u \end{bmatrix}, \ \boldsymbol{S} = \begin{bmatrix} 0 \\ -\rho\phi_x \\ -\rho u\phi_x \end{bmatrix},$$

where $\boldsymbol{U}$ is the vector of unknowns, $\boldsymbol{F}(\boldsymbol{U})$ is the flux-term, $\boldsymbol{S}$ shows corresponding source term, ρ is density,u denotes the particle velocity, P is pressure,ϕ is an time independent gravitational potential, and finally E is the total energy, which can be obtained as

$$E = \frac{P}{\gamma - 1} + \frac{1}{2}\rho u^2,$$

where γ represents the ratio of specific heat. The relevant eigenvalues and eigenvectors for the defined system of equations are expressed as

$$\lambda_1 = u - c, \ \lambda_2 = u, \ \lambda_3 = u + c.$$

$$\boldsymbol{r}_1 = \begin{bmatrix} 1 \\ u - c \\ \zeta - uc \end{bmatrix}, \ \boldsymbol{r}_2 = \begin{bmatrix} 1 \\ u \\ u^2/2 \end{bmatrix}, \ \boldsymbol{r}_3 = \begin{bmatrix} 1 \\ u + c \\ \zeta + uc \end{bmatrix},$$

where in the above equations c and ζ are called sound speed and the total specific enthalphy, respectively, which can be computed as

$$c = \sqrt{\frac{\gamma P}{\rho}}, \ \zeta = \frac{E+P}{\rho}.$$

In order to solve the above system of equation the wave propagation algorithm described in the next section is used.

3. Wave propagation algorithm

The one dimensional Godunov-type wave propagation algorithm can be given as [6, 7]

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^n - \frac{\triangle t}{\triangle x}(\boldsymbol{A}^+ \triangle \boldsymbol{U}_{i-1/2} + \boldsymbol{A}^- \triangle \boldsymbol{U}_{i+1/2}) - \frac{\triangle t}{\triangle x}(\tilde{\boldsymbol{F}}^n_{i+1/2} - \tilde{\boldsymbol{F}}^n_{i-1/2}),$$

where $\boldsymbol{U}^{n+1}$ is the vector of unknowns at the next time step, $\boldsymbol{A}^{+}\triangle\boldsymbol{U}_{i-1/2}$ and $\boldsymbol{A}^{-}\triangle\boldsymbol{U}_{i-1/2}$ provide the right- and left-going fluctuations, and finally $\tilde{\boldsymbol{F}}^{n}_{i\pm1}$ shows the second-order correction terms required to obtain high-resolution scheme, which can be used with different choice of limiters. If $\tilde{\boldsymbol{F}}=0$, the first-order Godunov-type wave propagation algorithm is achieved. The right and left-going fluctuations are then computed, using the following formulations

$$\boldsymbol{A}^{+}\triangle\boldsymbol{U}_{i-1/2}=\sum_{k:\lambda_{i-1/2>0}}\boldsymbol{\xi}_{k,i-1/2},\ \boldsymbol{A}^{-}\triangle\boldsymbol{U}_{i-1/2}=\sum_{k:\lambda_{i-1/2<0}}\boldsymbol{\xi}_{k,i-1/2},$$

where $\boldsymbol{\xi}_{k,i-1/2}$ is called the kth flux-wave propagating from cell interface $i-1/2$ and can be obtained by multiplying particular coefficients into its corresponding eigenvector, say,

$$\boldsymbol{\xi}_{k,i-1/2}=\beta_{k,i-1/2}\boldsymbol{r}_{k,i-1/2}.$$

4. Flux-wave formula

The flux-wave approach has been first introduced by [1] for the acoustic problem. This approach has been later modified by Mahdizadeh et al. [11,12] for the SWEs with modeling wet/dry front abilities. In this section this modified version of flux-wave approach is extended for the solution of one dimensional Euler equations with gravitational source term .The original flux-wave formula including the treatment of source term can take the form [1]

$$\boldsymbol{F}(\boldsymbol{U}_i)-\boldsymbol{F}(\boldsymbol{U}_{i-1})-\boldsymbol{S}_{i-1/2}\triangle x=\sum_{k=1}^{M_w}\boldsymbol{\xi}_{k,i-1/2},$$

where $\boldsymbol{F}(\boldsymbol{U}_i)$ and $\boldsymbol{F}(\boldsymbol{U}_{i-1})$ are the fluxes at the left and right side of cell interface $i-1/2$ and M_w denotes the number of waves, which for the prescribed Euler equations is equal to three and $\triangle x$ implies finite-volume cell length. To expand the flux-wave approach for the one dimensional Euler equations it is only required that the differences between neighboring fluxes minus the source terms is equalized with the summation of the relevant fluxes . This can be accomplished as

Woodward–Coella

$$\begin{bmatrix}\rho_i\tilde{u}_i\\ \rho_i\tilde{u}_i+P_i\\ (E_i+P_i)\tilde{u}_i\end{bmatrix}-\begin{bmatrix}\rho_{i-1}\tilde{u}_{i-1}\\ \rho_{i-1}\tilde{u}^2_{i-1}+P_{i-1}\\ (E_{i-1}+P_{i-1})\tilde{u}_{i-1}\end{bmatrix}-\triangle x\begin{bmatrix}0\\ -\rho_i(\phi_{i+1}-\phi_i)/\triangle x\\ -\rho_i u_i(\phi_{i+1}-\phi_i)/\triangle x\end{bmatrix}\quad(1)$$

$$= \beta_1 \begin{bmatrix} 1 \\ \tilde{u}_i - \tilde{c}_i \\ \tilde{\zeta}_i - \tilde{u}_i c_i \end{bmatrix} + \beta_2 \begin{bmatrix} 1 \\ \tilde{u}_i \\ \tilde{u}_i^2/2 \end{bmatrix} + \beta_3 \begin{bmatrix} 1 \\ \tilde{u}_i + \tilde{c}_i \\ \tilde{\zeta}_i + \tilde{u}_i c_i \end{bmatrix},$$

where $\tilde{u}$ and $\tilde{\zeta}$ are the velocity and total specific enthalpy again, which can be obtained through the combination of exact and approximate Riemann wave speeds fully explained in [11], where the approximate Riemann solver utilized herein is based upon the Roe solver [14] . For the Euler equations the approximate velocity and total specific enthalpy can be givens as

$$\tilde{u} = \frac{\sqrt{\rho_{i-1}}u_{i-1} + \sqrt{\rho_i}u_i}{\sqrt{\rho_{i-1}} + \sqrt{\rho_i}}, \quad \tilde{\zeta} = \frac{\sqrt{\rho_{i-1}}\zeta_{i-1} + \sqrt{\rho_i}\zeta_i}{\sqrt{\rho_{i-1}} + \sqrt{\rho_i}},$$

and the sound speed $\tilde{c_i}$ can take the form

$$\tilde{c_i} = \sqrt{(\gamma - 1)(\tilde{\zeta_i} - 1/2\tilde{u}_i^2)}.$$

The systems mentioned in equation (1) can be rewritten as

$$\begin{bmatrix} 1 & 1 & 1 \\ \tilde{u}_i - c_i & \tilde{u} & \tilde{u}_i + \tilde{c_i} \\ \tilde{\zeta}_i - \tilde{u}_i c_i & \tilde{u}_i^2/2 & \tilde{\zeta}_i + \tilde{u}_i c_i \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \end{bmatrix}, \tag{2}$$

where δ_1, δ_2, and δ_3 are

$$\begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \end{bmatrix} = \begin{bmatrix} \rho_i \tilde{u}_i - \rho_{i-1}\tilde{u}_{i-1} \\ (\rho_i \tilde{u}_i + P_i) - (\rho_{i-1}\tilde{u}_{i-1}^2 + P_{i-1}) + \rho_i(\phi_{i+1} - \phi_i) \\ (E_i + P_i)\tilde{u}_i - (E_{i-1} + P_{i-1})\tilde{u}_{i-1} + \rho_i u_i(\phi_{i+1} - \phi_i) \end{bmatrix}.$$

By solving the linear system provided in equation (2), the coefficients β_1, β_2, and β_3 are computed, which can be later used to calculate the flux-wave $\boldsymbol{\xi}_{k,i-1/2}$ required for obtaining the left- and right-going fluctuations for the first-order Godunov-type wave propagation algorithm. It should be stressed for the solution of linear system given above the LU decomposition algorithm with partial pivoting defined in [13] at each time-step.

4.1 Stability conditions

To ensure the method's stability, the Courant–Friedrichs–Lewy condition (CFL) [4] similar to the SWEs is used. This condition can be given as the following equation for the one dimensional Euler equations:

$$\text{CFL} = \frac{max(\tilde{\lambda})}{\triangle t} \triangle x,$$

where $\tilde{\lambda} = max(\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\lambda}_3)$ is the maximum amount of wave speeds, where each wave speed is obtained as

$$\tilde{\lambda}_1 = \tilde{u}_i - \tilde{c}_i,\ \tilde{\lambda}_2 = \tilde{u_i},\ \tilde{\lambda}_3 = \tilde{u_i} + \tilde{c}_i.$$

Generally, the wave propagation algorithm uses values of CFL number close to one, which ultimately reduces the computational setup time compared to other Riemann solvers provided in literature.

5. Numerical results

In order to examine the effectiveness of modified flux-wave approach (MFW) for the solution of the one dimensional Euler equations, in this section several numerical test cases are provided. The suitability of the proposed MFW approach in dealing with the gravitational source terms is first investigated, and the obtained numerical results are compared with both reference solutions and available numerical data borrowed from literature. Then, the well-balanced property of the defined method with the existence gravitational field for the isothermal equilibrium is verified, and the calculated numerical results are validated with the nonwell-balanced WENO scheme. Finally, the capability of the proposed well-balanced model in capturing small and large amount of amplitude perturbation is tested. It should be expressed that the MFW model defined herein was solved, using an in-house FORTRAN code on an Intel Core (i7-4790) 3.6 GHz processor with 16GB of RAM. Additionally, the solver employs high-resolution wave propagation algorithm based on the monotonized centered (MC) limiter. The number of computational finite volume cells is defined separately for each test-case.

5.1 Shock tube problem with gravitational field

The purpose of this test case is to assess the performance of the proposed numerical solver for the solution of the one dimensional Euler equations with the existence of gravitational source term. For this problem the Sod test case is considered with the initial condition as

$$(\rho^0, u^0, P^0) = \begin{cases} (1, 0, 1) & if\ x \leq 0.5, \\ (0.125, 0, 1) & otherwise, \end{cases}$$

where ρ^0, u^0, and P^0 are again the initial data for the defined Riemann problem at time $t = 0$. As the boundary conditions the extrapolation boundary conditions were imposed for both left and right boundaries. In order to consider the effect of gravitational source term a constant exhibit gravitational

field $\phi_x = 1$ is used within the source term. The computation is implemented until time $t = 0.2$s with 100 uniform cells and the CFL number equal to 0.9. Figure 2 shows numerical predictions for the density, pressure, energy and velocity obtained based on the MFW approach. The validations are performed with the reference solution achieved using 2000 numerical cell and also with a novel higher-order WENO scheme defined by [9] with the same computational volumes. For the density field a left-going rarefaction wave along with the contact discontinuity and shock wave are created within solution and the obtained results are in qualitative agreement with both reference solution and the WENO approach. Additionally, due to the effect of gravitational source term the density is pulled upward for the left boundary.

Figure 1(b) demonstrates the numerical results for the pressure field. As can be observed, a left-going rarefaction waves as well as right-moving shock are appeared within solution and again the agreement between the obtained numerical results and the reference solution are quite well. Figures 1(c) and 1(d) exhibit the numerical results for the energy and velocity, respectively. In terms of the energy field a rather similar shape to the pressure is seen. For the velocity distributions a negative velocities in some regions are appeared, which is mainly caused by gravitational term.

5.2 One dimensional gas-falling into fixed external potential

This test case was originally introduced in [15, 18] and utilized to verify the well-balanced property of the proposed MFW approach for the isothermal equilibrium within gravitational field. The initial conditions of the problem can be given as

$$\rho = \rho_0 \exp\left(-\frac{\phi}{RT}\right),\ u = 0, \text{ and } p = RT\left(-\frac{\phi}{RT}\right), \tag{3}$$

where gravitational field is expressed as

$$\phi(x) = -\phi_0 \frac{L}{2\pi} \sin\left(\frac{2\pi x}{L}\right),$$

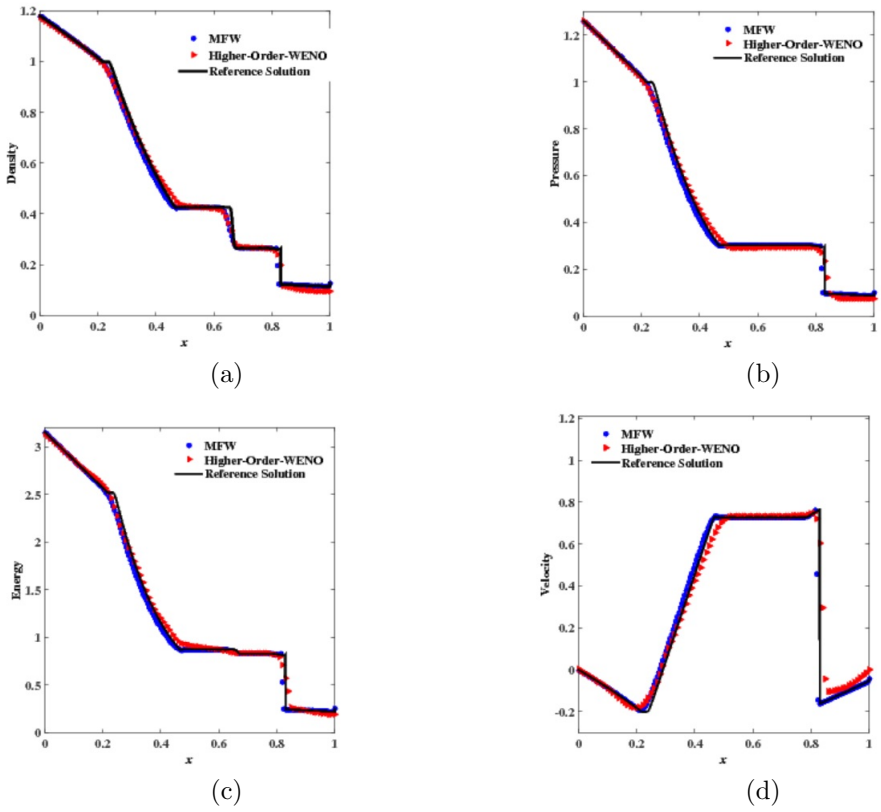


Figure 1: Numerical solution of the shock tube problem with gravitational source term performed at $t = 0.2s$. (a) Density, (b) pressure,(c) energy,(d) velocity

where$L$ is the computational domain set equal to 64 and other parameters are: $\rho_0 = 1$, $\gamma = 5/3$, $T = 0.6866$ and $\phi_0 = .02$. The solution was then performed in double precision using 200 and 400 computational cells until time t=50s, where the steady-state condition is reached. Table 1 demonstrates the Euclidean norm achieved for the conserved variables ρ, ρu and E at the steady-state condition. As can be seen for all variables relatively small amount of error is obtained, which clearly state that the initial conditions have been preserved during the simulations.

In order to compare the performance of the defined MFW approach with the unbalanced scheme a small perturbation was imposed into the steady-state solution provided in (3).Therefore the initial condition related to pressure becomes

$$p = RT\left(-\frac{\phi}{RT}\right) + 0.001\exp\left(-10(\mathrm{x} - 32)^2\right).$$

Table 1: Euclidean error norm computed based on the MFW approach for the steady-state

Number of Cells	ρ	ρu	E
200	2.6493E-04	9.7563E-05	4.3708E-04
400	6.8740E-05	2.2979E-05	8.9109E-05

The simulation was implemented up to time $t = 179980.53$s, which is much larger than the reference sound crossing time approximated as $\tau = 120$s. Figure 2 illustrates the numerical results obtained based upon the second-order MFW approach in comparison with non-well-balanced WENO scheme borrowed from [9]. As can be observed the initial condition for the velocity profile was maintained during computation in contrast to the non-well-balanced scheme, which was not able to properly neutralize the effect of gravitational source terms with the flux gradient for the isothermal equilibrium state.

5.3 Small and large amplitude wave propagation

The final test case employed herein was introduced in [5] and investigates the performance of the well-balanced MFW approach in dealing with small and large disturbances for the polytropic atmosphere in the gravitational field $\phi(x) = gx$. In doing so, the polytropic steady state solution is defined as

$$\rho(x) = \left(\rho_0^{\gamma-1} - \frac{1}{K_0}\frac{\gamma-1}{\gamma}gx\right)^{\frac{1}{\gamma-1}},\ u(x) = 0,\ p(x) = K_0\rho(x)^{\gamma}.$$

with the constants: $g = 1$, $\gamma = 5/3$, $\rho_0 = 1$, $p_0 = 1$, and $K_0 = p_0/(\rho_0^{\gamma})$. The computational domain of the polytropic atmosphere was set equal to $[0, 2]$ and the following periodic velocity perturbation was imposed at the bottom boundary

$$u(0, t) = A\sin(4\pi t), \tag{4}$$

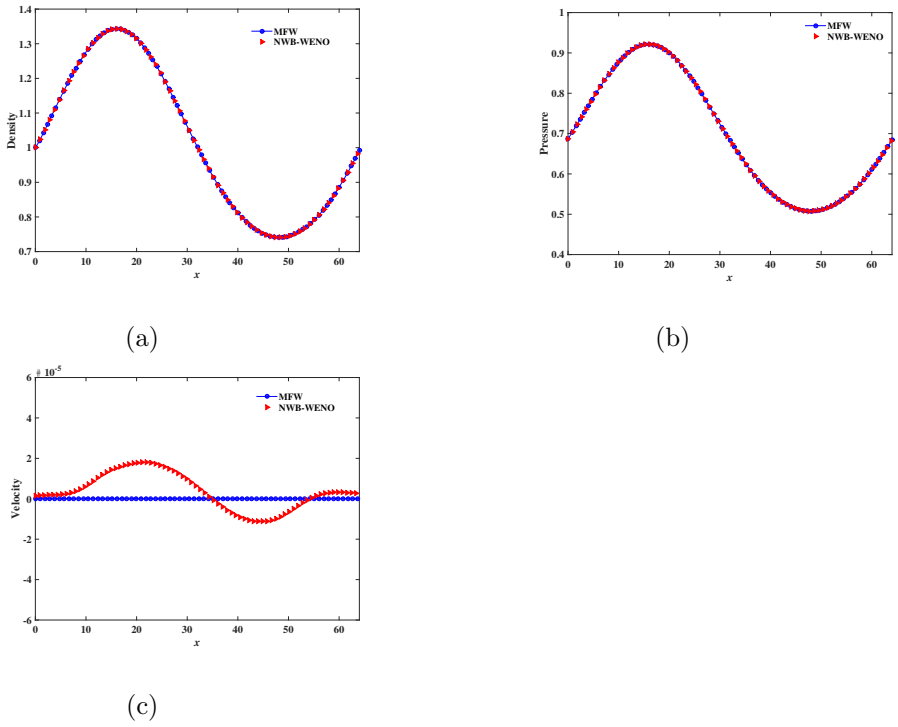
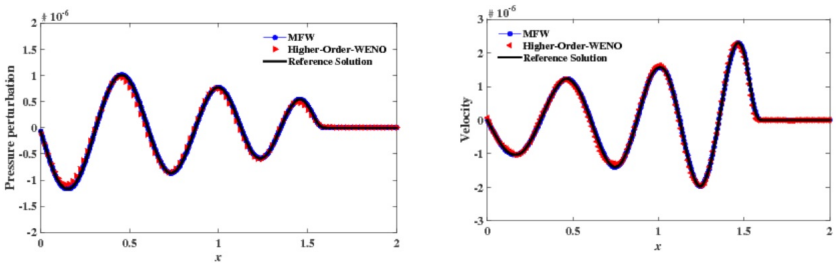


Figure 2: The numerical results achieved with the MFW method along with the non well-balanced WENO scheme for the perturbation problem given in section 5.2 at $t = 179.980.53$s. (a) Density, (b) pressure,(c) velocity

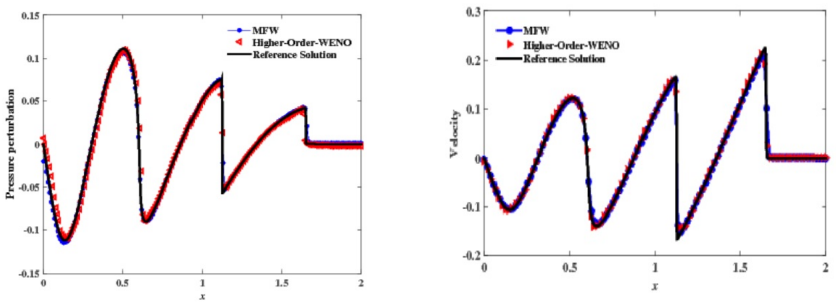
In the first numerical study the amount of A in above equation was chosen to 10^{-6}which provides a small perturbation into the solution and the simulation was run until time $t = 1.5$s. In Figure 2 the pressure perturbation and velocity profiles calculated based upon the second-order MFW approach with 400 numerical cells for the small disturbance were shown. Theses results have been also compared with the reference solution with 8000 numerical cells together with the higher-order WENO scheme given in [9]. It is clear that the results are in good agreement with both reference solution and the higher-order WENO scheme, which confirms that even for a small-perturbation the effect of gravitational source term has been accurately treated within the flux-differences of the neighboring cells for the finite volume method.



(a) (b)

Figure 3: Numerical results obtained based on the MFW approach for the small perturbation imposed for the polytropic steady-state solution in comparison with the higher-order WENO scheme and reference. solution (a) Pressure perturbation,(b) velocity

Eventually the amount of A in the equation (4) was set to 0.1, which creates rather large amplitude disturbance into the solution. The simulation based on the second-order MFW approach with 200 computational cells and CFL=0.9 was then implemented until t=1.5s . Figure 4 shows plots for the pressure perturbations and velocity along with the results calculated using reference solution with 2000 cells and higher-order WENO scheme. As can be observed the large amplitude perturbation was also captured by the defined MFW approach and the results are in qualitative agreement with the higher-order WENO method.



(a) (b)

Figure 4: Numerical results obtained based on the MFW approach for the large perturbation imposed for the polytropic steady-state solution in comparison with the higher-order WENO scheme and reference. solution (a) Pressure perturbation, (b) velocity

Conclusions

In this paper a modified flux-wave formula is presented for the solution one dimensional Euler equations including gravitational source term. The numerical solver defined herein is well-balanced and deals with gravitational source terms inside the differences between fluxes for the finite volume computational cells. In order to cope with difficulties with nonphysical results, a combination of exact and approximate Riemann solution was utilized. A modification for the Sod problem was first considered with the gravitational source term and the validations was made with the higher-order WENO scheme and rather identical results was achieved. Then, the well-balanced property of the scheme was tested and the ability of the method in modeling the perturbation applied into the steady-state equilibrium in comparison with unbalanced scheme was demonstrated. For the problem with small and large amplitude of perturbations, the MFW approach gave the same results equal to higher-order WENO scheme and accurately captured disturbances imposed to the polytropic atmosphere.

References

1. Bale, D. S., LeVeque, R. J., Mitran, S. and Rossmanith, J. A. *A wave propagation method for conservation laws and balance laws with spatially varying flux functions*, SIAM J. Sci. Comput. 24 (3) (2003), 955–978.

2. Botta, N., Klein, R., Langenberg, S.and Lutzenkirchen, S. *Well balanced finite volume methods for nearly hydrostatic flows*, J. Comput. Phys. 196 (2004), 539-565.

3. Chandrashekar, P. and Klingenberg, C. *A second order well-balanced finite volume scheme for euler equations with gravity* SIAM J. Sci. Comput. 37 (3) (2015) B382–B402.

4. Courant, R., Friedrichs, K. O. and Lewy, H. *Uber die partiellen differenzeengleichungen der mathematischen* physik. Mass.Ann. 100 (1928) 32–74.

5. KAppeli, R. and Mishra, S. *Well-balanced schemes for the euler equa-˜ tions with gravitation* J. Comput. Phys. 259 (Supplement C) (2014), 199–219.

6. LeVeque, R. J. *Balancing source terms and flux gradients in highresolution godunov methods: The quasi-steady wave-propagation algorithm* J. Comput. Phys. 146 (1998), 346–365.

7. LeVeque, R. J., *Finite volume methods for hyperbolic problems* Cambridge University Press, 2002.

8. LeVeque, R. J. and Bale, D. S. *Wave propagation methods for conservation laws with source terms* In: in Proceedings of the 7th International Conference on Hyperbolic Problems (1998), pp. 609–618.

9. Li, G. and Xing, Y.,*High order finite volume WENO schemes for the euler equations under gravitational fields* J. Comput. Phys. 316 (2016), 145–163.

10. Luo, J., Xu, K. and Liu, N. *A well-balanced symplecticity-preserving gas-kinetic scheme for hydrodynamic equations under gravitational field* SIAM J. Sci. Comput. 33 (5) (2011), 2356–2381.

11. Mahdizadeh, H., Stansby, P. K. and Rogers, B. D. *On the approximation of local efflux/influx bed discharge in the shallow water equations based on a wave propagation algorithm* Int J. Numer. Methods. Fluids. 66 (10) (2011), 1295–1314.

12. Mahdizadeh, H., Stansby, P. K. and Rogers,B. D. *Flood wave modeling based on a two-dimensional modified wave propagation algorithm coupled to a full-pipe network solver* J. Hydraul. Eng. 138 (3) (2012), 247–259.

13. Press, W. H., ATeukolsky, S., Vetterling, W. T. and Flannery, B. P. *Numerical Recipes in Fortran 77* Cambridge University Press, 1992.

14. Roe, P. *Approximate riemann solvers, parameter vectors, and difference schemes* J. Comput. Phys. 43 (2) (1981), 357–372.

15. Tian, C., Xu, K., Chan, K. and Deng, L. *A three-dimensional multidimensional gas-kinetic scheme for the navierˆastokes equations under gravitational fields* J. Comput. Phys. 226 (2) (2007), 2003– 2027.

16. Toro, E. F. *Riemann Solvers and Numerical Methods for Fluid Dynamics* Springer, 1997.

17. Toro, E. F., Castro, C. E. and Lee, B. J. *A novel numerical flux for the 3d euler equations with general equation of state* J. Comput. Phys. 303 (2015), 80–94.

18. Xing, Y. and Shu, C.-W. *High order well-balanced weno scheme for the gas dynamics equations under gravitational fields* J. Sci. Comput. 54 (2) (2013), 645–662.

Iranian Journal of Numerical Analysis and Optimization
Vol. 8, No. 2, (2018), pp 39–54
DOI:10.22067/ijnao.v8i2.61465

New variable neighborhood search method for minimum sum coloring problem on simple graphs

Kh. Erfani*, S. Rahimi and J. Fathali

Abstract

The minimum sum coloring problem (MSCP) is to find a legal vertex coloring for G using colors represented by natural numbers $(1, 2, \dots)$ such that the total sum of the colors assigned to the vertices is minimized. The aim of this paper is to present the skewed variable neighborhood search (SVNS) for this problem based on a new structure of neighborhoods. To increase the speed of the neighborhood search process, we present the new concepts of holder vertex and set. Tested on 23 commonly used benchmark instances, our algorithm shows acceptable competitive performance with respect to recently proposed heuristics.

Keywords: Minimum sum coloring; Variable neighborhood search; Skewed variable neighborhood search; Chromatic sum; Holder vertex; Holder set; Reducer set.

1 Introduction

The general graph coloring is one of the most well-known problems in combinatorial optimization. Besides its theoretical significance as a canonical NP-Hard problem [9], graph coloring arises naturally in a variety of real-world

*Corresponding author
Received 31 December 2016; revised 17 June 2017; accepted 28 February 2018
Kh. Erfani
Department of Applied Mathematics, Shahrood University of Technology, Shahrood, Iran.
e-mail: khalilerfani@gmail.com

S. Rahimi
Department of Applied Mathematics, Shahrood University of Technology, Shahrood, Iran.
e-mail: srahimi40@yahoo.co.uk

J. Fathali
Department of Applied Mathematics, Shahrood University of Technology, Shahrood, Iran.
e-mail: fathali@shahroodut.ac.ir

applications such as timetable problems [3], warehouse management [29], frequency allocation in mobile network [28], register allocation in optimizing compilers [4], scheduling problem [8], design and operation of exible manufacturing systems [10]. For example, in computer systems with multiprocessors that are in competition over resources, we might seek an allocation under which no two jobs with conflicting requirements are executed simultaneously while minimizing the average completion time of the jobs.

MSCP is closely related to the basic graph coloring problem. It was proposed by Kubicka [21] in the field of graph theory and by Supowit [30] in the field of VLSI design.

Suppose that $G = (V, E)$ is a simple undirected graph (without loop, multi, and directed edge) with vertex set $V = \{v_1, v_2, \ldots, v_n\}$ and edge set $E \subseteq V \times V$. A proper vertex k-coloring of G is an assignment of positive integers to its vertices by function $c : V \to \{1, 2, \ldots, k\}$ such that no two adjacent vertices are assigned the same number (i.e. $c(v_i) \neq c(v_j), \forall (i, j) \in E$). The color of a vertex v is denoted by $c(v)$.

A legal k-coloring can also be defined as partition of V in to k independent sets $V_1, V_2, \ldots, V_k$ such that for all $u, v \in V_i (i = 1, \ldots, k), (u, v) \notin E$. In other word, c can also be represented as a partition of V in to k mutually disjoint independent set (called color classes) $V_1, V_2, \ldots, V_k$ such that $\bigcup_{i=1}^{n} V_i = V$ and $v \in V_i$ if and only if $c(v) = i$.

The general graph coloring problem (GCP) is to determine a proper k-coloring with a minimum value of k. This value is called the chromatic number of G and denoted by $\chi(G)$. A related problem to the GCP is the minimum sum coloring problem (MSCP), which is to find a proper coloring $c = \{V_1, V_2, \ldots, V_k\}$ such that the following total sum of color labels, denoted by $\sum_c G$, to be minimized.

$$\sum_c G = \sum_{i=1}^{n} c(v_i) = \sum_{i=1}^{k} i|V_i|.$$

Now the vertex chromatic sum or minimum sum coloring of G is denoted by $\sum G$, and defined as

$$\min \left\{ \sum_c G \mid c \in C \right\},$$

where C is the collection of all proper coloring of G. The vertex strength of G denoted by $s(G)$, is the smallest number s, where there is a proper coloring c with s colors such that $\sum_c(G) = \sum G$. It is clear that $s(G)$ is lower bounded by $\chi(G)$. The distance between $s(G)$ and $\chi(G)$ can be large. Indeed there are many instances in which $s(G) >> \chi(G)$ [21].

As shown in [20, 21], the decision version of the MSCP is NP-complete in the general case. As a result, solving the MSCP is computationally challenging and any algorithm able to determine the optimal solution of the problem

is expected to require an exponential complexity. Also, it has several practical applications including VLSI design, scheduling, and distributed resource allocation (see [24] for a list of references for more application).

Due to its high computational complexity, no polynomial-time algorithm can solve or approximate the problem efficiently unless $P = NP$. In the past several decades, much effort has been devoted to developing various heuristic and metaheuristic algorithms. For the purpose of practical solving of the general MSCP, several heuristic algorithms have recently been proposed to find suboptimal solutions. This class of algorithms has been mainly developed since 2009 and is not proved the optimality of the solution found. Some examples of the heuristic algorithms include tabu search [2], greedy algorithms [23], genetic and memetic algorithms [5, 17–19, 26, 32], breakout local search [1], iterated local search [15] , ant colony [6] as well as heuristics based on independent set extraction [33, 34]. But in contrast, some kinds of literature have addressed the issue of theoretical bounds that formally proved, included [7, 11, 12, 16, 22, 31] for instance.

In recent years, variable neighborhood search (VNS) has been proven as a very effective and adaptable metaheuristic, used for solving a wide range of complex optimization problems. In this paper, we design and test a VNS algorithm for finding proper colorings, which correspond to upper bounds of the chromatic sum. In the literature on graph coloring several proposals appeared, combining the following characteristics: partial vs complete colorings, proper vs improper colorings, fixed vs variable k (see [15] for a review).

We opt vector X with size of $|V|$ for a solution representation consisting of colors that must be proper. We then devise a search strategy that borrows ideas from modified VNS method called skewed VNS method (SVNS) towards special kind of neighborhood. The fact that increasing the number of colors may decrease the chromatic sum was the main motivation of definition of such new neighborhood structure (see Fig. 1).

In the following sections, we first provide integer nonlinear mathematical formulation along with a definition of N_k neighborhood structure of MSCP. In the next section, we offer a new concept called *holding* to expedite the related local search method. In Section 4 we present our SVNS method with respect to the neighbors that are mentioned in Section 2. Before concluding, detailed computational results and comparisons with five algorithms are presented in Section 5.

2 Modeling and N_k Neighborhoods of MSCP

2.1 Modeling

Let $G = (V, E)$ be a simple undirected graph with vertex set $V = \{v_1, v_2, \ldots, v_n\}$ and edge set $E \subseteq V \times V$. Although the MSCP can be formulated as a binary quadratic problem, we also presented another formulation of this problem. Define variable x_i as the color of vertex i for $i = 1, 2, \ldots, n$. Therefore the integer nonlinear mathematical formulation is defined as follows:

$$\begin{aligned} Min \quad & \sum_{i=1}^{n} x_i \\ s.t. \quad & \\ & |x_i - x_j| \geq 1 \qquad : \forall (i,j) \in E \qquad (1) \\ & 1 \leq x_i \leq n \quad , integer \quad : i = 1, 2, \ldots, n \qquad (2) \end{aligned}$$

The previous model can be reduced with elimination of redundant constraints and suitable changing of variables,

$$\begin{aligned} P) \quad Min \quad & \sum_{i=1}^{n} x_i \qquad (3) \\ s.t. \quad & \\ & |x_i - x_j| \geq 1 \qquad : \forall (i,j) \in E \qquad (4) \\ & x_i \geq 0, integer \quad : i = 1, 2, \ldots, n \qquad (5) \end{aligned}$$

This mathematical formulation expresses MSCP correctly, since each vector $X = (x_1, x_2, \ldots, x_n)$ that satisfies equations (4) and (5) is a proper coloring c for G and the objective function (3) minimizes $\sum_c G$. $\sum(X)$ denotes the sum of colors of proper coloring c. It should be noted that this model does not claim about $s(G)$ and focuses on $\sum G$. By assuming that S be set of all proper coloring of G, P can be written briefly as follows:

$$\begin{aligned} P) \quad Min \quad & \overrightarrow{1}.X \qquad (6) \\ s.t. \quad & \\ & X \in S, \qquad (7) \end{aligned}$$

where $\overrightarrow{1}.X$ is the inner product of $\overrightarrow{1} = (1, 1, \ldots, 1)$ and X. Indeed $\sum(X) = \sum_{i=1}^{n} x_i = \overrightarrow{1}.X$. According to the description stated above, any feasible solution for P is a vector $X = (x_1, x_2, \ldots, x_n)$ satisfied equations (4) and (5). So the next neighborhoods should be defined around this vector.

2.2 New Neighborhoods for MSCP

We know that the neighborhood is an important element that influences the local search procedure. So, searching for feasible spaces is meaningless without knowing the definition of the neighborhood. Therefore according to the mathematical model of the previous part, we introduce neighboring structure for MSCP here. To read more about neighborhoods for this problem you can see [1, 14, 15, 17].

Definition 1. Let X and S be a feasible solution and feasible space of P, respectively. For $k = 0, 1, \ldots, n$, a neighborhoods of X are defined by

$$N_k(X) = \{Y \in S \mid Y \ \textit{is greater than } X \textit{ in at most } k \textit{ element.}\} \quad (8)$$

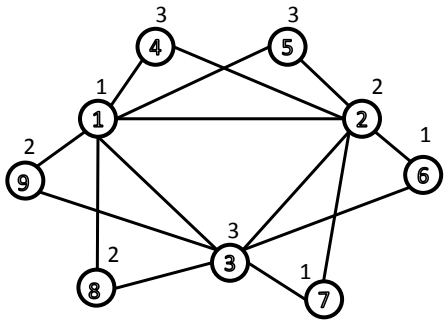
In fact, $N_k(X)$ contains all of the feasible solutions (proper coloring) that are greater than X in the k element at most. The size of the $N_k(.)$ becomes bigger if k increases. Indeed $|N_k(.)| \in O(\binom{n}{k})$ and we have

$$N_0(X) \subseteq N_1(X) \subseteq \cdots \subseteq N_k(X)$$

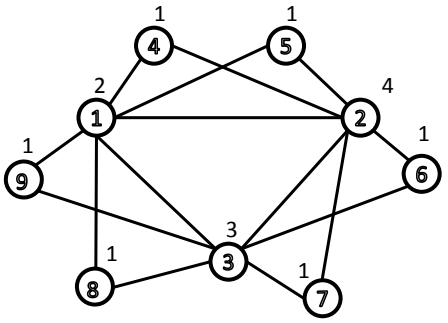
As you can see, these neighborhoods will be nested; that is, each one contains the previous. So if X is locally optimal with respect to N_k, then it will be locally optimal with respect to N_i for $i = 0, 1, \ldots, k-1$. Also, if it is not to be locally optimal w. r. t. N_i, then it will not be locally optimal w. r. t. N_i for $i = k+1, \ldots, n$. Clearly $S = N_n(X)$. But it's important to know the minimum value of k such that $N_k = S$. In other words, for which value of k the neighborhood N_k is exact? For a given neighborhood N if any local optimal is also global optimal, then N is exact [27].

To clarify the issue, Figure 1 shows an illustrative example for $N_k(.)$. The numbers that are in and out the vertices are vertex number and vertex color, respectively. With the given 3-coloring X (left figure), we achieve the suboptimal sum of 18 while Y leads to chromatic sum of 15 (right figure). By Definition 1, $N_0(X) = \{X\}$ and $Y \notin N_0(X), N_1(X)$ but $Y \in N_k(X) for all k \geq 2$. Therefore by starting with X, we achieve the locally optimal Y w. r. t. N_2, that is also globally optimal.

The next important point that should be noted is about the size of N_k. As stated previously by increasing k the size of neighborhood becomes large and this causes the process of searching happens slowly. Therefore we will offer a new concept called *holding* and try to expedite to find the local optimum with this new concept.



(a) $X = (1, 2, 3, 3, 3, 1, 1, 2, 2)$



(b) $Y = (2, 4, 3, 1, 1, 1, 1, 1, 1)$

Figure 1: An illustrative example for $N_k(.)$

3 Holder and Reducer Set

Definition 2. ***k-holder set:*** Given a proper coloring $X = (x_1, \ldots, x_n)$ and k-element subset $H = \{t_1, t_2, \ldots, t_k\}$ from V. H is a k-holder set, when for some $v \in N(H)$[1] and index $0 \leq r \leq k$ the following two conditions hold:

1) $x_{t_r} < x_v$;

2) For any vertex $w \in N(v) \setminus H$, we have $x_w \neq x_{t_r}$.

In such case, we say that H holds vertex t_r.

Especially in above definition when $k = 1$, we obtain 1-holder set called *holder vertex*. Indeed, for two adjacent vertices a and b, we say a holds b and denoted by $a \nearrow b$, if Definition 1 is stated for $k = 1$. Consider again the graph of Figure 1a with 3-coloring X. Both two sets $H_1 = \{1\}$ and $H_2 = \{2\}$ are holder vertices according to Definition 2. H_1 is a *good holder vertex* because H_1 holds $4, 5, 9, 8$ and when its color is increased the objective function

[1] Denote the set of adjacent vertices of H

is reduced three units. In contrast H_2 is not a good one because no change does happen when its color starts to increase. Thus, the holding feature not sufficient to improve the objective function and should have *reduction* property.

Definition 3. ***k-reducer set:*** Given a proper coloring X. The k-holder set H is a k-reducer set, when the objective function is improved (reduced) by increasing its color of vertices.

The above definitions lead us to the following obvious propositions, which more specifies the relationship between N_k and these two new concepts.

Proposition 1. *The proper coloring X is locally optimal w.r.t. N_k if and only if G has no k-reducer set.*

Proposition 2. *The proper coloring X is a global optimal if and only if there exists $k_0 \in \mathbb{Z}^+$ such that G has no k-reducer set w.r.t. X for any $k \geq k_0$.*

For a given solution X, if G has no k-holder set for any $k \in \mathbb{Z}^+$, then it has no k-reducer set for any $k \in \mathbb{Z}^+$ and so according to Proposition 2 X is globally optimal.

4 VNS Method with respect to N_k for MSCP

Variable neighborhood search (VNS) algorithm was originally described by Mladenovic and Hansen ([13,25]). The basic strategy of the VNS is to focus the investigation of the solutions, which belong to some neighborhood of the current best one. In order to avoid being trapped in local suboptimal solutions, VNS changes the neighborhoods, directing the search in the promising and unexplored areas. By this systematic change of neighborhoods, VNS iteratively examines a sequence of neighbors of the current best solution. But, it may happen that some instances have several separated and possibly far apart local optimum containing near-optimal solutions. If one considers larger and larger neighborhoods, the information related to the currently best local optimum dissolves. It is therefore of interest to modify VNS schemes in order to explore more fully local optima, which are far away from the current solution. This will be done by accepting to recenter the search when a solution close to the best one known, but not necessarily as good, is found, provided that it is far from this last solution. So, we study on the modified and special ingredient of VNS called skewed variable neighborhood search (SVNS) that is presented in Algorithm 1. The relaxed rule for recentering uses an evaluation function linear in the distance from the current solution; that is, $\sum(X'')$ is replaced by $\sum(X'') - \alpha\rho(X, X'')$ (line 13 of Algorithm 1). where $\rho(X, X'')$ is the distance from X to X'' and α a parameter that is determined experimentally. A metric for distance between solutions is usually

Algorithm 1 Skewed variable neighborhood search for MSCP

Adjacency matrix of graph G , Select the set of neighborhood structures $N_k , k = 1, \ldots, k_{max}$. Find an initial proper coloring X and its value $\sum(X)$. Set $X_{opt} \longleftarrow X$. Choose a stopping condition and a parameter value α. A solution X_{opt} and its value $\sum(X_{opt})$. stopping condition not reached Set: $k \longleftarrow 1 \;\; k \leq k_{max}$ Generate a solution X' at random from the k^{th} neighborhood of X or perturb X, randomly./**Shaking or Perturbing* */ Apply some local search method with X' as initial solution. Denote with X'' the so obtained local optimum. /* *Local search**/ $\sum(X'') < \sum(X_{opt})$ $Xopt \longleftarrow X'' \sum(X_{opt}) \longleftarrow \sum(X'')$ /**Improvement or not**/ $\sum(X'') - \alpha\rho(X, X'') < \sum(X)$ $X \longleftarrow X'' \; k \longleftarrow 1 \; k \leftarrow k + 1$ /**Move or not.**/

easy to find, for example, the hamming distance when solutions are described by boolean vectors or the Euclidean distance in the continuous case. Here we use p-norm distance for p=1,2. P-norm is a class of vector norms, denoted by $||.||_p$, is defined as

$$||X||_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{\frac{1}{p}} \quad p \geq 1, X \in \mathbb{R}^n$$

and the distance between to two n-vectors X and Y based on this norm, is defined as

$$||(X, Y)||_p = (|x_1 - y_1|^p + |x_2 - y_2|^p + \cdots + |x_n - y_n|^p)^{\frac{1}{p}}$$

Algorithm1 presents the general SVNS algorithm for the MSCP, whose ingredients are detailed in the following. It has four main phases, *shaking or perturbing, local search, improve or not* and *move or not.* Usually, the second phase will have maximum computing time. SVNS starts with an initial random coloring X, which must be proper. In the first phase (line 4) we use shaking or perturbing at random. Indeed, we decide shaking to shake against perturbing strategy with probability p in each iteration of local search procedure. k_{max}, the maximum value for the size of neighborhoods, must be identified at first. In local search phase (line 6), we apply variable neighborhood descent (VND) as an ingredient of VNS method. This method is illustrated in Algorithm 2. Then SVNS check to ensure that any improvement has happened or not (lines 8–11)? In lines 13–18 it decides on moving to the new solution or increasing k. SVNS iterates these steps while the maximum number of iterations since the last improvement is reached.

As noted previously, the maximum computing time is spent on the local search phase. To improve the solution quality and acceleration of local search, we will use the holding concept described in Section 2. With determining the suitable *order of falling*, the exploration velocity of neighbors is increased. A

perturbation mechanism will be used to improve solution quality and jumping from local optimums (attractors).

Algorithm 2 Variable neighborhood descent

Select the set of neighborhood structures $N_l\,, l = 1, \ldots, l_{max}$. Find an initial proper coloring X and its value $\sum(X)$. A solution X and its value $\sum(X)$. there is improvement Set $l \longleftarrow 1 \;\; l \leq l_{max}$ Find the best neighbor X' of X $(X' \in N_l(X))$./*Exploration of Neighborhood*/ $\sum(X') < \sum(X) \;\; X \leftarrow X' \; l \longleftarrow 1 \; l \leftarrow l + 1$ /*Move or not*/

4.1 Order of Falling

According to Section 2, the existence of k-holder sets are a *necessary condition* for the existence of k-reducer sets. Therefore, we will use this necessary condition for recognition of reducer sets. Then we increase all color of vertices in reducer set for *falling* the colors of remaining vertices. It is important that the remaining vertices meet *in what order* to take maximum falling of color vertices happens.

Suppose graph G with proper coloring X. Construct weighted directed graph $D = (V(D), E(D), W)$ from G as follows:

1) $V(D) = V(G)$.

2) $E(D) = \{(a, b) |\, a, b \in V(D)\, \&\, a \nearrow b\}$.

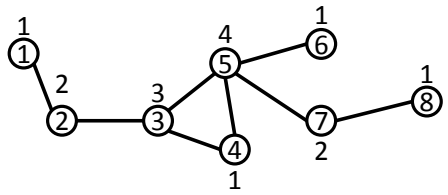
3) $\forall\, e = (a, b) \in E(D)$, $w_e = x_b - x_a$.

D is a weighted directed graph with no directed cycle, multiple edge, and loop. Actually, D is a directed acyclic graph (DAG). The longest directed path in D can help us to find the best order of falling.

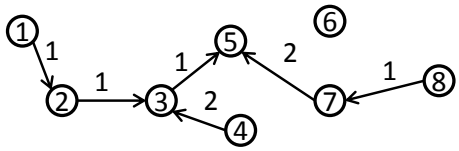
Although the longest path problem is NP-hard for a general graph, it has a linear time solution for directed acyclic graphs. The idea is based on topological sorting. Topological sorting for a DAG is a linear ordering of vertices such that for every directed edge (u, v), vertex u comes before v in the ordering. Topological sorting for a graph is not possible if the graph is not a DAG.

Figure 2 shows an illustrative example of the order of falling. Figure 2a shows G with proper 4-coloring X with sum 15 and corresponding weighted DAG D is depicted in Figure 2b. D has three directed longest paths $p_1 = 1, 2, 3, 5$, $p_2 = 4, 3, 5$ and $p_3 = 8, 7, 5$ of length 3. Therefore they are examples of order of falling. For p_1 it means that if the first vertex of p_1 (vertex 1) is omitted (or increase its color), then the vertex colors of $2, 3, 5$ begin to decrease, sequentially. Finally, the color of vertex 1 re-determined and

another proper 3-coloring $Y = (2, 1, 2, 1, 3, 1, 2, 1)$ with sum 13 is obtained from X. In fact, $Y \in N_1(X)$. Now we can apply the above process for new solution Y.
With this idea, the examination of all elements of N_k to find a local optimum of N_k is not necessary and therefore the exploration velocity of neighbors in local search phase of Algorithm 1 increases.



(a) G with $X = (1, 2, 3, 1, 4, 1, 2, 1)$



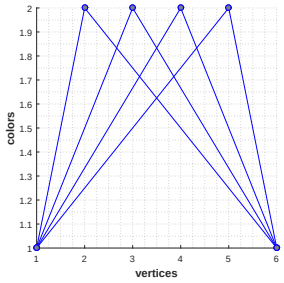
(b) The weighted DAG D

Figure 2: An illustrative example for the order of falling

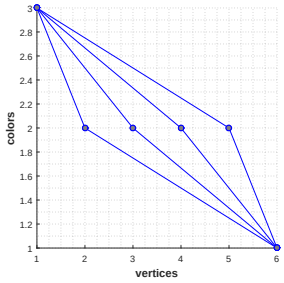
4.2 Perturbation Mechanism

Consider the sub optimal solution X, which has not any suitable falling order. Indeed the weighted DAG D is an empty or low-density graph. Increasing the value of k usually is not useful and lead to increase the computation time. In these cases, local search procedure may be induced to cycle between two or more locally optimal solutions and leading to search stagnation. So, to jump from this situations we change in X to create some good falling orders. We will choose vertex (or vertices) that has more interruptions for holding of the other vertices and increase their colors to produce holder vertices and consequently generate orders of falling to decrease objective function.

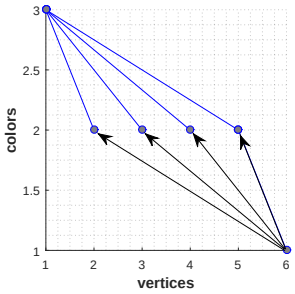
We explain this mechanism on the example shown in Figure 3. We have plotted this example on two-dimensional axis vertex-color. The horizontal and vertical axises are the index and color of vertices respectively. Suppose $k_{max} = 1$, and we stop in proper coloring $X = (1, 2, 2, 2, 2, 1)$ (Fig.3a). There is no holder set of size one and consequently, there is no one-reducer set according to X. Therefore Proposition 1 implies that X is locally optimal (but not globally) with respect to N_1. To jump from this situation we should gen-



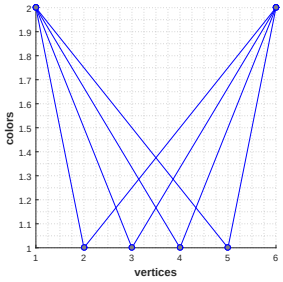
(a) Local optimal that isn't global



(b) Increasing the color of vertex 3



(c) Vertex 6 becomes a holder vertex



(d) Global optimal

Figure 3: An illustrative example for perturbation mechanism

erate holder set by increasing the color of some vertices. But which vertices? Increasing color of one or even all 2,3,4, and 5 causes nothing in order to improve the solution. But increasing color of vertex 1 (or 6) is very useful because it causes to generate holder and reducer set $H = \{6\}$(Figs.3c,3b). If now we apply local search w.r.t N_1 on new solution $Y = (3, 2, 2, 2, 2, 1)$, then we get the new better proper coloring $(2, 1, 1, 1, 1, 2)$, which is globally optimal (Fig.3d).

5 Experimental Results

Our SVNS algorithm is programmed in *MATLAB(R2015a)* and compiled on a PC with 2.7 GHz CPU and 4 Gb RAM. The 23 of well-known benchmark graphs from *COLOR*[2] $2002 - 2004$ in the literature, which is commonly used to test sum coloring algorithms is considered in Table 1 and Table 2. The

[2] http://mat.gsia.cmu.edu/COLOR02/

reported values are based on 10 independent runs. Table 1 gives the detailed characteristics of these benchmark graphs. Columns 1–6 indicate the number of vertices ($|V|$), the number of edges ($|E|$), the density $d = \frac{2mn}{n-1}$, the chromatic sum ($\sum$) and the smallest number of required colors (k) of graphs. Columns 7–9 present detailed computational results of our SVNS algorithm: Best result obtained ($\sum_*$) with the number of required colors (k_*), average coloring sum (*Avg.*) and standard deviation (*Std.*). The results, for instances, of COLOR02 benchmark indicate that SVNS attained the best-known result for 19 instances, and was unable to reach the current best result for four instances *(homer, queen8.8, miles250 and miles500)*. Table 2 reports the

Table 1: Computational results of SVNS on 23 COLOR02 instances

	Characteristics of graphs					SVNS		
Instances	$\|V\|$	$\|E\|$	d	$\sum$	k	$\sum_*(k_*)$	Avg.	Std.
myciel3	11	20	0.36	21	4	21 (4)	21.0	0.0
myciel4	23	71	0.28	45	5	45 (5)	45.0	0.0
myciel5	47	236	0.22	93	6	93 (6)	93.0	0.0
myciel6	95	755	0.17	189	7	189 (7)	189.0	0.0
myciel7	191	2360	0.13	381	8	381 (8)	381.4	0.52
anna	138	986	0.05	276	11	276 (11)	276.3	0.48
david	87	812	0.11	237	11	237 (11)	238.6	0.84
huck	74	602	0.11	243	11	243 (11)	243.0	0.0
jean	80	508	0.08	217	10	217 (10)	217.1	0.31
homer	561	1629	0.01	-	10	1163 (13)	1168.5	3.5
queen5.5	25	160	0.53	75	5	75 (5)	75.0	0.0
queen6.6	36	290	0.46	138	7	138 (8)	138.2	0.42
queen7.7	49	476	0.4	196	7	196 (7)	197.4	4.4
queen8.8	64	728	0.36	291	9	294 (9)	301	5.33
games120	120	638	0.09	443	9	443 (9)	445.3	1.56
miles250	128	387	0.05	325	8	329 (8)	333.2	2.1
miles500	128	1170	0.11	$\leq$709	20	719(20)	728.2	7.03
mug88-1	88	146	0.04	178	4	178 (4)	178.0	0.0
mug88-25	88	146	0.04	178	4	178 (4)	178.0	0.0
mug100-1	100	166	0.03	202	4	202 (4)	202.0	0.0
mug100-25	100	166	0.03	202	4	202 (4)	202.0	0.0
2-insertion-3	37	72	0.11	62	4	62 (4)	62.0	0.0
3-insertion-3	56	110	0.07	92	4	92 (4)	92.0	0.0

comparative results with the following approaches from the literature on the tested COLOR02 instances: a heuristic EXSCOL [33], a greedy algorithm (MRLF) based on the popular RLF graph coloring heuristic [23], MDS5 [15], a parallel genetic algorithm (PGA) [19], a hybrid local search (HLS) [5]. The comparisons are based on the criterion of quality; that is, the smallest sum of colors reached by a given algorithm. Notice that information like computing

Table 2: Comparative results between our SVNS algorithm and five reference approaches on the set of COLOR02 instances

Names	$\sum$	EXCOL [33]	MRLF [23]	MDS5 [15]	PGA [19]	HLS [5]	SVNS
myciel3	21	21	21	21	21	21	**21**
myciel4	45	45	45	45	45	45	**45**
myciel5	93	93	93	93	93	93	**93**
myciel6	189	189	189	189	189	189	**189**
myciel7	381	381	381	381	382	381	**381**
anna	276	283	277	276	281	-	**276**
david	237	237	241	237	243	-	**237**
huck	243	243	244	243	243	243	**243**
jean	217	217	217	217	217	-	**217**
queen5.5	75	75	75	75	75	-	**75**
queen6.6	150	138	138	138	138	138	**138**
queen7.7	196	196	196	196	196	-	**196**
queen8.8	291	291	303	291	302	-	294
games120	443	443	446	443	460	446	**443**
miles250	325	328	334	325	347	343	329
miles500	$\leq$709	709	715	712	762	755	719
mug88-1	178	-	-	178	-	-	**178**
mug88-25	178	-	-	178	-	-	**178**
mug100-1	202	-	-	202	-	-	**202**
mug100-25	202	-	-	202	-	-	**202**
2-insertion-3	62	-	-	62	-	-	**62**
3-insertion-3	92	-	-	92	-	-	**92**

time are not available for all the reference algorithms. Also the symbol "-" means that the information is not available.

6 Conclusion

In this paper, we have presented the skewed variable neighborhood search algorithm towards new kind of neighborhood for solving the minimum sum coloring problem. To acceleration, we introduce a new holding concept in graph coloring problems too. The computational evaluation of the proposed algorithm on 23 of COLOR02 benchmark instances has revealed that SVNS attain the best-known results for 19 instances while failing to reach the best ones for four instances. Also, these results have acceptable competitive compared with five algorithms for MSCP in final.

Acknowledgements

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

References

1. Benlic, U. and Hao, J. K. *A study of breakout local search for the minimum sum coloring problem*, pp. 128–137, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

2. Bouziri, H. and Jouini, M. *A tabu search approach for the sum coloring problem*, Electronic Notes in Discrete Math. 36 (2010) 915–922.

3. Burke, E. K., McCollum, B., Meisels, A., Petrovic, S. and Qu, R. *A graph-based hyper-heuristic for educational timetabling problems*, Eur. J. Oper. Res. 176 (2007), no. 1, 177 – 192.

4. de Werra, D., Eisenbeis, Ch.,Lelait, S. and Marmol, B.*On a graph- theoretical model for cyclic register allocation*, Discrete Appl. Math. 93 (1999), no. 23, 191 – 203.

5. Douiri S.M. and Elbernoussi, S. *New algorithm for the sum coloring problem*, Int. J. Contemporary Math. Sci. 6 (2011) no. 9-12, 181–192.

6. Douiri S. M. and Elbernoussi, S. *A new ant colony optimization algorithm for the lower bound of sum coloring problem*, J. Math. Model. Algorithms 11 (2012), no. 2, 181–192.

7. Erdos, P., Kubicka, E. and Schwenk, A. J. *Graphs that require many colors to achieve their chromatic sum*, In Proceedings of the Twentieth South-eastern Conference on Combinatorics Graph Theory and Computing (Boca Raton, FL), vol. 71, 1990, pp. 17–28.

8. Gamache, M., Hertz, A. and Ouellet, J. O. *A graph coloring model for a feasibility problem in monthly crew scheduling with preferential bidding*, Comput. Oper. Res. 34 (2007), no. 8, 2384–2395.

9. Garey, M. R. and Johnson, D. S. *Computers and intractability; a guide to the theory of np-completeness*, W. H. Freeman & Co., New York, NY, USA, 1990.

10. Glass, C. A. *Bag rationalisation for a food manufacturer*, J. Oper. Res. Soc. 53 (2002), no. 5, 544–551.

11. Hajiabolhassan, H.,Mehrabadi, M.L. and Tusserkani, R. *Minimal coloring and strength of graphs*, Discrete Math. 215 (2000), no. 1, 265 – 270.

12. Hajiabolhassan, H.,Mehrabadi, M. L. and Tusserkani, R. *Tabular graphs and chromatic sum*, Discrete Math. 304 (2005), no. 1-3, 11–22.

13. Hansen, P. and Mladenovic, N. *Variable neighborhood search: Principles and applications*, Eur. J. Oper. Res. 130 (2001) no. 3, 449–467.

14. Hao J. -K. and Wu, Q. *Improving the extraction and expansion method for large graph coloring*, Discrete Appl. Math. 160 (2012), no. 1617, 2397 – 2407.

15. Helmar, A. and Chiarandini, M. *A local search heuristic for chromatic sum*, Proceedings of the 9th Metaheuristics International Conference, MIC 2011 (L. D. Gaspero, A. Schaerf, and T. Stutzle, eds.), 2011, pp. 161–170.

16. Jiang, T. and West, D. B. *Coloring of trees with minimum sum of colors*, J. Graph Theory 32 (1999), no. 4, 354–358.

17. Jin, Y.,Hao, J. K. and Hamiez, J. P. *A memetic algorithm for the minimum sum coloring problem*, Comput. Oper. Res. 43 (2014), 318 – 327.

18. Jin, Y. and Hao, J. -K. *Hybrid evolutionary search for the minimum sum coloring problem of graphs*, Inf. Sci. 352 (2016), no. C, 15–34.

19. Kokosiński Z. and Kwarciany, K. *On sum coloring of graphs with parallel genetic algorithms*, pp. 211–219, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

20. Kroon, L. G., Sen, A. , Deng, H. and Roy, A. *The optimal cost chromatic partition problem for trees and interval graphs*, pp. 279–292, Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

21. Kubicka, E. *The chromatic sum of a graph*, Ph.D. thesis, Western Michigan University, Michigan, 1989.

22. Kubicka, E. and Schwenk, A. J. *An introduction to chromatic sums*, Proceedings of the 17th Conference on ACM Annual Computer Science Conference (New York, NY, USA), CSC '89, ACM, 1989, pp. 39–45.

23. Li, Y. ,Lucet, C., Moukrim, A. andSghiouer, K. *Greedy Algorithms for the Minimum Sum Coloring Problem*, Logistique et transports (Sousse, Tunisia), March 2009, pp. LT–027.

24. Malafiejski, M. *Sum coloring of graphs*, pp. 55–65, AMS, Poland, 2004.

25. Mladenovic, N. and Hansen, P. *Variable neighborhood search*, Comput. Oper. Res. 24 (1997) no.11, 1097–1100.

26. Moukrim, A., Sghiouer, K., Lucet, C. and Li, Y., *Lower bounds for the minimal sum coloring problem*, Electronic Notes in Discrete Math. 36 (2010), 663 – 670.

27. Papadimitriou, C. H. and Steiglitz, K. *Combinatorial optimization: Algorithms and complexity*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1982.

28. Smith, D.H., Hurley, S. and Thiel, S.U. *Improving heuristics for the frequency assignment problem*, Eur. J. Oper. Res. 107 (1998), no. 1, 76 – 86.

29. Stecke, K. E. *Design, planning, scheduling, and control problems of exible manufacturing systems*, Ann. Oper. Res. 3 (1985), no. 1, 1–12.

30. Supowit, K. J. *Finding a maximum planar subset of a set of nets in a channel*, Trans. Comp.-Aided Des. Integ. Cir. Sys. 6 (2006), no. 1, 93–94.

31. Thomassen, C., Erdos, P. , Alavi, Y., Malde, P. J. and Schwenk, A. J. *Tight bounds on the chromatic sum of a connected graph*, J. Graph Theory 13 (1989) no. 3, 353–357.

32. Wang, Y., Hao, J. K.,Glover, F. and Lu, Zh. *Solving the minimum sum coloring problem via binary quadratic programming*, CoRR abs/1304.5876 (2013).

33. Wu Q. and Hao, J. -K. *An effective heuristic algorithm for sum coloring of graphs*, Comput. Oper. Res. 39 (2012), no. 7, 1593 – 1600.

34. Wu Q. and Hao, J. -K. *Improved lower bounds for sum coloring via clique decomposition*, CoRR abs/1303.6761 (2013).

Iranian Journal of Numerical Analysis and Optimization
Vol. 8, No. 2, (2018), pp 55–74
DOI:10.22067/ijnao.v8i2.51969

Technological returns to scale: Identification and visualization

E. Hajinezhad and M.R. Alirezaee*

Abstract

One of the most critical issues for using data envelopment analysis models is the identification of technological returns to scale (TRTS). Recently, the angles method based on data mining is introduced for the identification of TRTS. This objective method uses the angles to measure the gap between the constant and variable TRTS. The gap is calculated in both the increasing and decreasing sections of the frontier. The larger the gap in the increasing and/or decreasing sections of the frontier, the closer TRTS is to the increasing and/or decreasing form of TRTS. In this paper, we propose a heuristic method for visualizing TRTS that would give a better understanding of identification of TRTS in the dataset. To this end, we introduce the maximum angles method for measuring the maximum possible deviation from constant TRTS assumption in the increasing and decreasing sections of the frontier. By the angles and the maximum angles , we can display the dataset's TRTS in a two-dimensional space. To validate the proposed method, we consider six one input/one output cases. Also, we apply the angles method and the maximum angles method for the Maskan bank of Iran. Using the proposed method, we show that how TRTS of the bank dataset can be displayed in a two-dimensional space.

Keywords: Data envelopment analysis; Returns to scale; Technology; Bank.

1 Introduction

Data envelopment analysis (DEA) is a nonparametric method for evaluating the efficiency of decision making units (DMUs) while each DMU utilizes

*Corresponding author
Received 18 December 2015; revised 31 January 2018; accepted 25 April 2018
E. Hajinezhad
Department of Applied Mathematics, School of Mathematics, Iran University of Science and Technology, Hengam St., Resalat, Tehran, Iran. e-mail: e.hajinezhad@iust.ac.ir

M.R. Alirezaee
Department of Applied Mathematics, School of Mathematics, Iran University of Science and Technology, Hengam St., Resalat, Tehran, Iran. e-mail: mralirez@iust.ac.ir

multiple inputs to produce multiple outputs. DEA is introduced by the CCR model [11], for which technological returns to scale (TRTS) is assumed to be constant. In fact, it is assumed that the technological behavior of the dataset is constant from returns to scale point of view, and we call it constant technological returns to scale (CTRS). In another words, by multiplying inputs with α ($\alpha \geq 0$), outputs are also multiplied by the same value. Removing this full proportionality assumption, results in the BCC model [5] with variable technological returns to scale (VTRS). Overall, different assumptions on the proportionality between the inputs and outputs lead to various TRTS. The determination of returns to scale (RTS) is done at two levels: Technology and DMU. When RTS is identified at the technological level, it can be determined at the DMU level. The relation between these two levels is as follows:

- If TRTS is identified to be constant, then RTS of every DMUs is constant.
- If TRTS is identified to be increasing, then RTS of a DMU might be constant or increasing.
- If TRTS is identified to be decreasing, then RTS of a DMU might be constant or decreasing.
- If TRTS is identified to be variable, then RTS of a DMU might be constant or increasing or decreasing.

For the determination of RTS at the DMU level, basic methods have developed in two main paths [6]: One of the paths, which was developed by Färe, Grosskopf, and Lovell [13,14], determines RTS by using ratio of radial measures. Model pairs are used for the development of these ratios and they only differ in satisfaction of convexity and sub-convexity conditions. The second path developed by Banker [3], Banker and Thrall [9], includes radial measure models as well as the additive and multiplicative models. Furthermore, many papers have focused on the determination of DMUs' RTS in different DEA models such as nonradial measure DEA models [17, 26], weight restricted DEA models [16, 18, 25], and FDH models [15, 19, 21]. Moreover, the measurement of RTS is considered in the presence of undesirable outputs [22, 23] and also negative data [2].

One of the most important issues in evaluating the efficiency of DMUs is the proper identification of RTS at the technological level. There are two basic approaches for the identification of TRTS: The subjective and objective approaches. Most of the objective methods are based on the statistics. The fundamental papers in this area are [4, 7, 8] in which the identification of TRTS is addressed by using DEA based hypothesis tests. Also, Simar and Wilson [20] have investigated TRTS identification by using nonparametric statistical tests. Using a statistical method mostly needs some assumptions to be made. Obviously, the wrong assumptions can lead to incorrect results. Although, these methods are strong from theoretical point of view, they may

be difficult to use. Due to these deficiencies, Alirezaee, Hajinezhad, and Paradi [1] recently proposed a novel nonstatistical method for the objective identification of TRTS, which is called the angles method.

The angles method has been developed based on data mining for measuring RTS at the technological level. To achieve this aim, the gap between two assumptions; that is, constant and variable TRTS, is measured by an angle. For each DMU, a hyperplane with VTRS assumption and a hyperplane with CTRS assumption are constructed; the angle between these two hyperplanes is calculated as the DMU's gap. Based on the layered RTS [1] of the DMU under study, the calculated angle is stored in the increasing or decreasing set. Eventually, the angles of the increasing set are aggregated into an angle, which shows the gap from the constant form of TRTS in the increasing section of the frontier. Similarly, the aggregation of the angles in the decreasing set represents the deviation from CTRS in the decreasing section of the frontier. The larger the angles in the increasing and/or decreasing sections of the frontier, the closer TRTS will be to the increasing and/or decreasing assumptions.

In this paper, we propose a heuristic method for visualizing TRTS in a two-dimensional space. To this end, we introduce the maximum angles method to show how far a frontier with VTRS assumption may deviate from CTRS assumption. The angle between the efficient frontiers with CTRS assumption and the weak efficient frontier with VTRS assumption, is considered as a candidate for the maximum angle. For creating the weak efficient frontiers, anchor points have been used. Finally, the maximum angles are calculated in the increasing and decreasing sections of the frontier. By the angles, maximum angles and the percentages of DMUs in each section of the frontier, the dataset's TRTS in one input/one output space is displayed. The proposed method is validated by 6 cases of one input/one output. Application of the angles method and the maximum angles method has been investigated on an Iranian bank dataset.

2 The angles method

In this section, the main points of the angles method introduced in [1], is briefly described. The main concept of this method is the gap, which shows the deviation from CTRS assumption. The gap is defined to show moving towards or away from the constant form of TRTS assumption. In this method, the gap is measured by the angle between two frontiers with different TRTS. For each DMU, a frontier with VTRS assumption and another one with CTRS assumption are constructed. Then the angle between these two frontiers is calculated.

In the angles method, RTS of the technology is determined by using all the DMUs' angles , because the technology influences all the DMUs' behavior. To

consider all the DMUs for determining TRTS, the layering technique [12,24] is applied. Let us consider N DMUs that use m inputs to produce s outputs in set PPS^1. In the layering technique, the first efficient layer is constructed by the efficient DMUs obtained from the execution of DEA model for PPS^1. By deleting the efficient DMUs from set PPS^1, the new set of DMUs PPS^2 is created. Again, the DEA model is executed for the DMUs of PPS^2 and the obtained efficient DMUs make up the second layer. In the same way, PPS^t is constructed by removing the DMUs on the layer $(t-1)$ from the set PPS^{t-1}; the layer t is formed by running the DEA model for the set PPS^t. This process continues until the number of DMUs is meaningful regarding the number of inputs and outputs. So, the layering continues until the number of DMUs is more than or equal to 3×(number of inputs + number of outputs). The DEA model used for the layering technique is the additive model with VTRS assumption.

After measuring a DMU's angle, it is required to determine whether it shows the movement to the increasing or decreasing TRTS assumption. To this end, the angles method uses RTS at the DMU level measured by the method presented in [6] with a little change. At iteration t of the layering, RTS of DMUs constructed the layer t, are measured regarding the DMUs in the set PPS^t. Since a DMU's RTS is determined based on the layer located on it, this RTS is called the layered RTS (LRTS). Assume that in iteration t of the layering, DMU j_o is under study and relies on layer t. Let j_o consume $x_o = (x_{1o}, \ldots, x_{mo})$ to produce $y_o = (y_{1o}, \ldots, y_{so})$. Moreover, let PPS^t be the set containing all the observed DMUs at iteration t. DMU j_o is evaluated by the following multiplier form of input oriented BCC model (it is clear that since j_o is strong efficient, it is efficient by the input oriented model as well):

$$
\begin{aligned}
Max \quad & uy_o - u_0 && && (1a)\\
s.t. \quad & uy_j - vx_j - u_0 \le 0, && j \in PPS^t, && (1b)\\
& vx_o = 1, && && (1c)\\
& u \ge 0,\ v \ge 0,\ u_0\ free\ in\ sign. && && (1d)
\end{aligned}
$$

Let us suppose that the optimum solution obtained from the above model is in the form of (u^*, v^*, u_0^*). If $u_0^* = 0$, the DMU's LRTS is constant. Otherwise if $u_0^* > 0$ $(u_0^* < 0)$, the following minimization (maximization) model for the LRTS identification is applied:

$$
\begin{aligned}
Min(Max) \quad & u_0 && && (2a)\\
s.t. \quad & uy_j - vx_j - u_0 \leq 0, && j \in PPS^t, && (2b)\\
& vx_o = 1, && && (2c)\\
& uy_o - u_0 = 1, && && (2d)\\
& u \geq 0,\ v \geq 0, && && (2e)\\
& u_0 \geq 0\ (u_0 \leq 0). && && (2f)
\end{aligned}
$$

If the optimized u_0 from the above model equals zero, the LRTS of the DMU under evaluation is constant. If u_0 is strictly positive (negative), its LRTS is decreasing (increasing).

Similarly, DMU j_o can be evaluated by the following multiplier form of output oriented BCC model (it is clear that since j_o is strong efficient, it is efficient by the output oriented model as well).

$$
\begin{aligned}
\min \quad & vx_o + v_0 && && (3a)\\
s.t. \quad & uy_j - vx_j - v_0 \leq 0, && j \in PPS^t, && (3b)\\
& uy_o = 1, && && (3c)\\
& u \geq 0,\ v \geq 0,\ v_0\ free\ in\ sign. && && (3d)
\end{aligned}
$$

Let us suppose that the optimum solution obtained from the above model is in the form of (u^*, v^*, v_0^*). If $v_0^* = 0$, the DMU's LRTS is constant. Otherwise if $v_0^* > 0$ ($v_0^* < 0$), the following minimization (maximization) model for the LRTS identification is applied:

$$
\begin{aligned}
\min(\max) \quad & v_0 && && (4a)\\
s.t. \quad & uy_j - vx_j - v_0 \leq 0, && j \in PPS^t, && (4b)\\
& uy_o = 1, && && (4c)\\
& vx_o + v_0 = 1, && && (4d)\\
& u \geq 0,\ v \geq 0, && && (4e)\\
& v_0 \geq 0\ (v_0 \leq 0). && && (4f)
\end{aligned}
$$

If the optimized v_0 from the above model equals zero, the LRTS of the DMU under study is constant. If v_0 is strictly positive (negative), its LRTS is decreasing (increasing).

To determine the angle corresponding to DMU j_o, two hyperplanes with constant and variable TRTS are required. Since j_o is a strong efficient DMU, it is efficient by the BCC model. So, the BCC hyperplane passing through j_o is considered as the hyperplane with VTRS assumption. However, it is possible that j_o is not CCR efficient. Therefore, the CCR hyperplanes passing through one of the reference DMUs for j_o from the additive model with CTRS assumption, is considered as the hyperplane with CTRS assumption. The set of mentioned reference DMUs is represented by Ref_{jo}. An impor-

tant point is that the BCC and CCR hyperplanes satisfying those conditions are almost infinite. Therefore, for having unique hyperplanes and avoiding overestimation of the angle between the hyperplanes, the hyperplanes with the smallest angle are selected. So, to determine the angle for a DMU, the smallest angle model is introduced [1]. Suppose that DMU j_o is on layer t. Since j_o is strong efficient regarding the set PPS^t, it is efficient by the input oriented BCC model, too. So, the angle corresponding to DMU j_o is evaluated by using the following input oriented smallest angle model:

$$\max \quad \frac{(v^c,u^c)^T(v^v,u^v)}{||(v^c,u^c)||\ ||(v^v,u^v)||} \tag{5a}$$

$$s.t. \quad v^c x_{j_R} = 1, \tag{5b}$$

$$u^c y_{j_R} = 1, \tag{5c}$$

$$-v^c x_j + u^c y_j \le 0 \qquad \forall j \in PPS^t, \tag{5d}$$

$$v^c \ge 0,\ u^c \ge 0, \tag{5e}$$

$$v^v y_o - u_0^* = 1, \tag{5f}$$

$$u^v x_o = 1, \tag{5g}$$

$$-v^v x_j + u^v y_j - u_0^* \le 0 \qquad \forall j \in PPS^t, \tag{5h}$$

$$v^v \ge 0,\ u^v \ge 0, \tag{5i}$$

where $v^c = (v_1^c, \ldots, v_m^c)$ and $u^c = (u_1^c, \ldots, u_s^c)$ are the input and output weights corresponding to the CCR model, $v^v = (v_1^v, \ldots, v_m^v)$ and $u^v = (u_1^v, \ldots, u_s^v)$ are the input and output weights corresponding to the BCC model, and u_0^* is the optimum value of u_0 from model (2). Maximizing the objective function (5a) —which is the inner product of the hyperplanes— leads to the hyperplanes with the smallest angle. Satisfying conditions (5b)–(5e) results in passing the hyperplane (v^c, u^c) through j_R –which is a member of the set Ref_{jo}– and supporting PPS^t with CTRS assumption. Similarly, satisfying conditions (5f)–(5i) results in passing the hyperplane (v^v, u^v, u_0^*) through j_o and supporting PPS^t with VTRS assumption. For every $j_R \in Ref_{jo}$, model (5) is executed and the smallest angle is selected as the angle corresponding to DMU j_o. If the DMU's LRTS is increasing, then the angle is saved in the increasing set represented by $Minangles^{ITRS}$. If the DMU's LRTS is decreasing, then the angle is stored in the decreasing set represented by $Minangles^{DTRS}$. ITRS and DTRS are the abbreviations for the increasing and decreasing technological returns to scale, respectively.

Similarly, it is assumed that DMU j_o is on the layer t. DMU j_o is efficient by the output oriented BCC model regarding the set PPS^t, since it is efficient by the additive model. So, the angle corresponding to DMU j_o can be evaluated by using the following output oriented smallest angle model:

$$\max \quad \frac{(v^c, u^c)^T (v^v, u^v)}{||(v^c, u^c)||\ ||(v^v, u^v)||} \tag{6a}$$

$$s.t. \quad v^c x_{j_R} = 1, \tag{6b}$$

$$u^c y_{j_R} = 1, \tag{6c}$$

$$-v^c x_j + u^c y_j \leq 0 \qquad \forall j \in PPS^t, \tag{6d}$$

$$v^c \geq 0,\ u^c \geq 0, \tag{6e}$$

$$v^v x_o + v_0^* = 1, \tag{6f}$$

$$u^v y_o = 1, \tag{6g}$$

$$-v^v x_j + u^v y_j - v_0^* \leq 0 \qquad \forall j \in PPS^t, \tag{6h}$$

$$v^v \geq 0,\ u^v \geq 0, \tag{6i}$$

where v_0^* is the optimum value of v_0 from model (4).

After finishing the layering process, the mean and median of the angles in each set; that is, $Minangles^{ITRS}$ and $Minangles^{DTRS}$, are calculated for determining the general gaps from CTRS assumption in the increasing and decreasing sections of the frontier. The geometric mean of the mean and median in the increasing section of the frontier; that is, $Mean^{ITRS}$ and Med^{ITRS}, is calculated as follows:

$$GM^{ITRS} = \sqrt{Mean^{ITRS} \times\ Med^{ITRS}}. \tag{7}$$

Moreover, the geometric mean of the mean and median in the decreasing section of the frontier; that is, $Mean^{DTRS}$ and Med^{DTRS}, is calculated as follows:

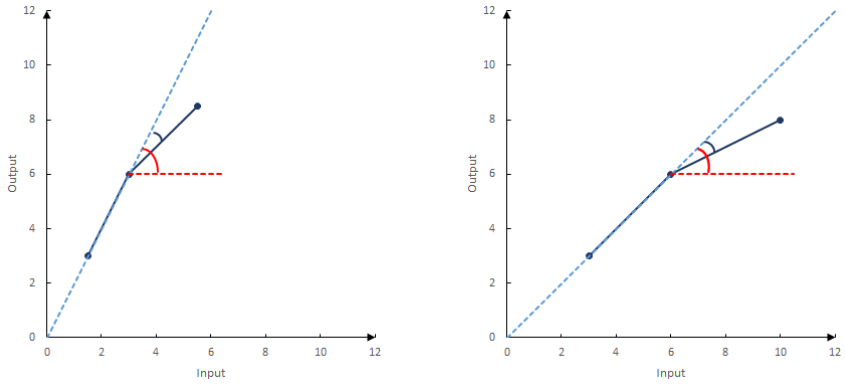
$$GM^{DTRS} = \sqrt{Mean^{DTRS} \times\ Med^{DTRS}} \tag{8}$$

GM^{ITRS} and GM^{DTRS} are also called the GM values in the increasing and decreasing sections of the frontier, respectively. GM^{ITRS} and GM^{DTRS} determine the tendency of the DMUs toward ITRS and DTRS assumptions.

3 A heuristic method for displaying TRTS

By using the angles method on a dataset, the angles in the increasing and decreasing sections of the frontier is calculated. However, these angles do not give any information about the maximum deviations from CTRS assumption. For clarification of this problem, consider two datasets in Figure 1. For both the datasets, the angle corresponding to the increasing section of the frontier is 18°. However, the maximum angles of case Figure 1a and Figure 1b are 63° and 45°, respectively.

Thus, for having a better picture of TRTS, the maximum angles in the increasing and decreasing sections of the frontier is required to be calculated. By having the maximum angles, we can represent a dataset's TRTS with



(a) (b)

Figure 1: The angles from CTRS assumption obtained by the angles method give no information about the maximum deviations

multiple inputs and multiple outputs in a two-dimensional space. In the following, we introduce a method that we call it the maximum angles method and it is used for calculating the maximum possible angles from CTRS assumption.

3.1 The maximum angles method

The weak efficient frontiers with VTRS assumption possess the maximum deviations from CTRS assumption. So, the maximum angle from CTRS assumption corresponds to a DMU lying on a weak efficient frontier. It means that the angles corresponding to the DMUs locating on the weak efficient frontiers need to be examined.

In order to calculate the maximum deviations for each layer in the increasing and decreasing sections of the frontier, it is necessary to examine all the weak efficient frontiers corresponding to that layer. On the other hand, it is possible that the weak efficient DMUs do not generate all the weak efficient frontiers. Therefore, we consider some artificial DMUs, which are weak efficient and produce all the weak efficient frontiers. To construct these artificial DMUs, the anchor points are used.

The intersections of the weak and strong efficient frontiers are called the anchor points [10]. In other words, a strong efficient DMU is an anchor point if and only if it is located on an unbounded hyperplane or a weak efficient frontier. By using anchor points, we create new DMUs on the weak efficient frontier and their corresponding angles are studied as the candidates

for the maximum angles in both sections of the frontier; that is, increasing and decreasing.

Consider the layer t. It can be easily shown that $\{e_1, e_2, \ldots, e_m\}$ and $\{-e_1, -e_2, \ldots, -e_s\}$ are the extreme directions of the production possibility set. e_i is a zero vector, where component i equals one. Suppose that j_o with (x_o, y_o) is a strong efficient DMU on the layer t. We construct $m+s$ artificial DMUs: Artificial DMU ${j_o}^i$ $(i = 1, \ldots, m)$ with $(x_o + e_i, y_o)$ and artificial DMU ${j_o}^r$ $(r = 1, \ldots, s)$ with $(x_o, y_o + 0.1 y_o e_r)$. To construct the artificial DMU ${j_o}^r$, the output of DMU j_o decreases proportionally to prevent negative outputs. If at least one of $m+s$ artificial DMUs is weak efficient, then DMU j_o is an anchor point on the layer t. However, the efficiency of all artificial DMUs are measured. Then, the angles corresponding to the weak efficient DMUs are calculated.

In a loop, artificial DMU ${j_o}^i$ $(i = 1, \ldots, m)$ is inserted to the set PPS^t (PPS^t is the set containing all the observed DMUs at iteration t of the layering) and evaluated by using the output oriented BCC model. Let artificial DMU ${j_o}^\iota$ with $(x_o + e_\iota, y_o)$ be a weak efficient DMU. Since, DMU j_o^ι is output oriented BCC efficient, its corresponding angle is calculated by using the output oriented smallest angle (6). If the DMU's LRTS is increasing (decreasing), then the smallest angle is stored in the set $MaxSet_t^{ITRS}$ ($MaxSet_t^{DTRS}$). Eventually, DMU j_o^ι is removed from the set PPS^t. Similarly, for every output component r $(r = 1, \ldots, s)$, artificial DMU j_o^r is inserted with $(x_o, y_o + 0.1 y_o e_r)$ to the set PPS^t and evaluated by using the input oriented BCC model. Suppose that DMU j_o^ρ is founded to be efficient. Since, DMU j_o^ρ is input oriented BCC efficient, its corresponding angle is calculated by using the input oriented smallest angle (5). If the DMU's LRTS is increasing (decreasing), then the smallest angle is stored in the set $MaxSet_t^{ITRS}$ ($MaxSet_t^{DTRS}$). Finally, the artificial DMU is removed from the set PPS^t. After the termination of both loops, the maximum members of the sets $MaxSet_t^{ITRS}$ and $MaxSet_t^{DTRS}$ are stored in Max_t^{ITRS} and Max_t^{DTRS}. Now, we can calculate the maximum angles in the two sections of the frontier by using equations (9)–(10).

The maximum GM values are calculated similar to the GM values. The mean and median of Max_t^{ITRS} for all layers, are saved in $Mean^{ITRS-M}$ and Med^{ITRS-M}. The maximum GM value in the increasing section of the frontier is calculated as follows:

$$GM^{ITRS-M} = \sqrt{Mean^{ITRS-M} \times Med^{ITRS-M}} \tag{9}$$

Moreover, The mean and median of Max_t^{DTRS} for all layers, are saved in $Mean^{DTRS-M}$ and Med^{DTRS-M}. The maximum GM value in the decreasing section of the frontier is calculated as follows:

$$GM^{DTRS-M} = \sqrt{Mean^{DTRS-M} \times Med^{DTRS-M}} \tag{10}$$

3.2 Displaying TRTS in the two sections of the frontier: increasing and decreasing

In this section, we want to introduce a heuristic method for displaying a dataset's TRTS in a two-dimensional space by using the results of the angles method as well as the maximum angles method. It should be noted that plotting the production possibility set (PPS) even in $(m + s =)$ three-dimensional space is difficult; so it is not a proper method for representing the dataset's TRTS. On the other hand, the scattering of data makes the PPS inappropriate for giving a clear image of TRTS.

By using the angles method, two angles and the percentage of DMUs in the increasing and decreasing sections of the frontier are obtained. By using the maximum angles method described in section 3.1, the maximum deviations from CTRS are calculated in the two sections of the frontier. These results are independent of the number of inputs and outputs; so it is assumed that these results are obtained for a one input/one output dataset. The angles are displayed by using a CCR and a BCC frontier. We explain the method of displaying TRTS on a case study, which is completely described in section 4.2.

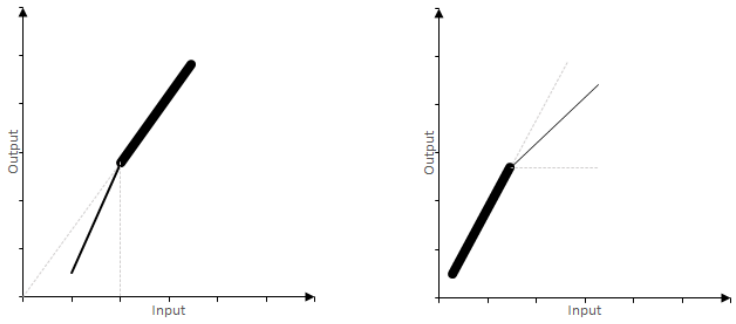
Consider the bank dataset (management code 256) with 22 DMUs that use two inputs to produce three outputs. The results of the angles method and the maximum angles method are represented in Table 1 . The values of GM and GM^{-M} represent the angle and the maximum possible angle from CTRS assumption, respectively. Also, Num represents the percentage of DMUs in both sections of the frontier.

Table 1: The results of the angles method and the maximum angles method for a bank dataset with 22 DMUs

Increasing section of the frontier			Decreasing section of the frontier		
$Num(\%)$	$GM(^\circ)$	$GM^{-M}(^\circ)$	$Num(\%)$	$GM(^\circ)$	$GM^{-M}(^\circ)$
22.2	11.7	35.7	11.1	18.0	61.6

By considering the results represented in Table 1, TRTS is displayed in Figure 2 according to the following rules:

- The CCR and BCC frontiers are represented by two lines.
- The CCR frontier is drawn in such a way that the angle between the CCR frontier and the weak efficient frontier in the increasing/decreasing section is equal to the maximum possible angle in those sections. This weak efficient frontier in the increasing and decreasing sections of the frontier are respectively, the vertical and horizontal lines represented by the dotted line.



(a) (b)

Figure 2: Displaying TRTS in the increasing (a) and decreasing (b) sections of the frontier for a bank dataset with 22 DMUs

- The percentage of DMUs in the constant section of the frontier is equal to 100 minus the total percentage of DMUs in the increasing and decreasing sections of the frontier.
- The higher the percentage of DMUs in the constant section of the frontier is the thicker CCR line .
- The BCC frontier in the increasing/decreasing section is drawn in such a way that the angle between the BCC frontier and the CCR frontier is equal to GM value in the increasing/decreasing section of the frontier.
- The higher the percentage of DMUs in the increasing/decreasing section of the frontier is the thicker BCC line .

The angles in the increasing and decreasing sections are represented separately because the maximum angles in both sections of the frontier may not be complementary; therefore the corresponding CCR frontiers do not coincide.

4 Experimental results

To study the results of the angles method and the maximum angles method, 6 cases one input/one output and a real dataset are considered. For each case, the GM values (by using equations (7)–(8)), the percentages of the DMUs and the maximum GM values (by using equations (9)–(10)) in the increasing and decreasing sections of the frontier are calculated.

To analyze the results of the angles method and the maximum angles method, the followings should be considered:

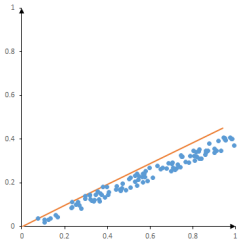
- The existence of small angles in the increasing and decreasing sections of the frontier is normal. Therefore, the angles smaller than 10° are insignificant and do not conflict with CTRS assumption.
- Large GM value in the increasing section of the frontier and small GM value in the decreasing section of the frontier, support ITRS assumption to be accepted.
- Small GM value in the increasing section of the frontier and large GM value in the decreasing section of the frontier, support DTRS assumption to be accepted.
- Large GM value in both sections of the frontier, supports VTRS assumption to be accepted.
- Large GM value produced by a small number of DMUs in the increasing and/or decreasing sections of the frontier, cannot reject CTRS assumption. So, if the number of DMUs in a section of the frontier is less than 10% of its total then the angle of that section is neglected.
- The experiments show clearly that the angles method does not merely accept or reject ITRS, CTRS or DTRS technology, but it determines the rate of increasing/decreasing if ITRS/DTRS is revealed. A larger GM value resulted in a larger rate of increasing/decreasing TRTS.
- It is obvious that a larger sample size would lead to more reliable results.

4.1 One input/one output samples

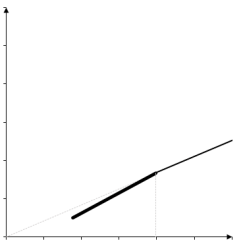
To study the results of the angles method and the maximum angles method, 6 one input/one output cases are considered. The cases are from [1] and plotted in the first column of Figure 3. TRTS of each case can easily be identified from Figure 3. TRTS of cases 3a–3c are constant. Also, TRTS of cases 3d, 3(e), and 3(f) are decreasing, increasing, and variable, respectively. For each cases, the angles method and the maximum angles method are applied. The results shown in Table 2, represent the percentage of the DMUs (Num), the GM values (GM), and the maximum GM values (GM^{-M}) in both sections of the frontier.

Regarding Table 2, we would make some points:

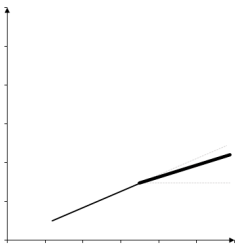
- For all the cases, the percentage of DMUs in both sections of the frontier is more than 17%. So, no angles are negligible.
- From Figure 3, it can be seen that the value of GM^{-M} in the decreasing section of the frontier is approximately equal to the angle between the CCR frontier and the horizontal line. Also, the value of GM^{-M} in the



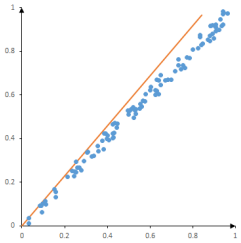
(a)



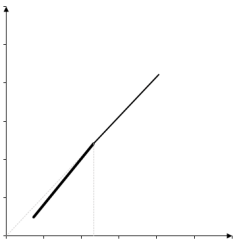
a - Increasing section



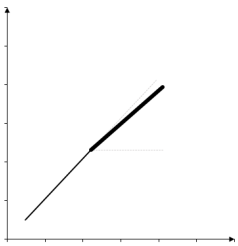
a - Decreasing section



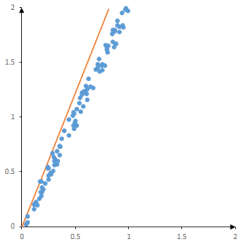
(b)



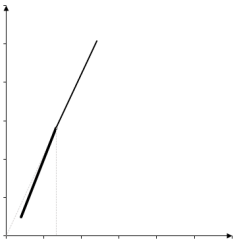
b - Increasing section



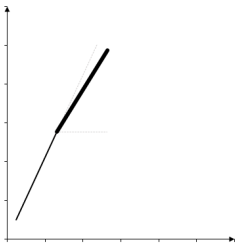
b - Decreasing section



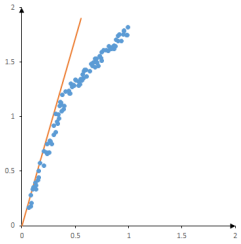
(c)



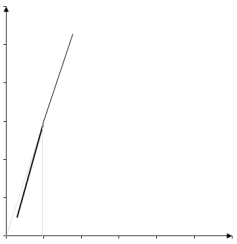
c - Increasing section



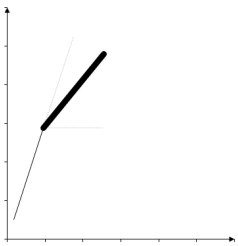
c - Decreasing section



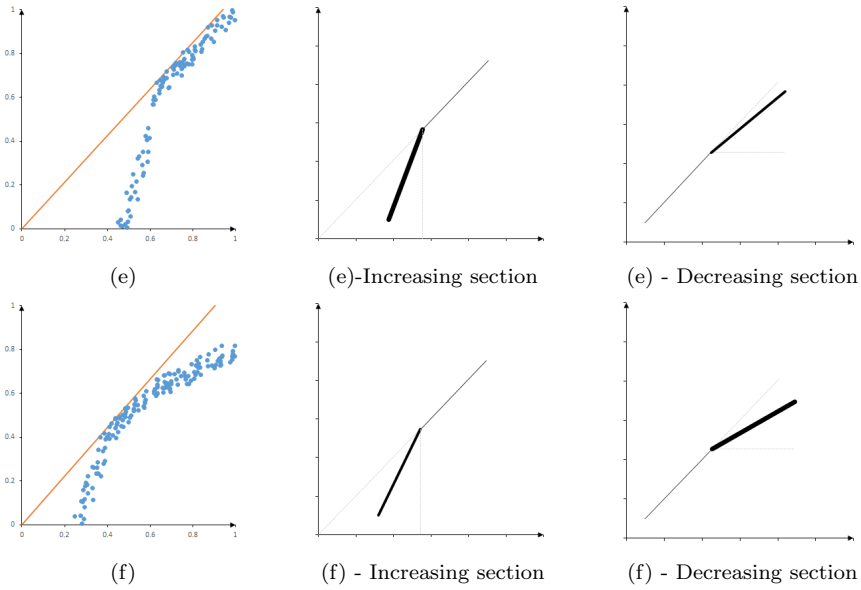
(d)



d - Increasing section



d - Decreasing section



(e) (e)-Increasing section (e) - Decreasing section

(f) (f) - Increasing section (f) - Decreasing section

Figure 3: Displaying the original one input/one output cases and their TRTS in the increasing and the decreasing sections of the frontier. In each row, the first cell represents the original one input/one output case [1], the second and the third cells display TRTS in the increasing and the decreasing sections of the frontier for the case in the first cell. The red line (the first column) shows the CCR frontier for the first layer. The horizontal and the vertical coordinates are the input and output, respectively

Table 2: The results of the angles method and the maximum angles method for one input/one output cases

Case	DMUs	Increasing section			Decreasing section		
		Num (%)	GM (°)	GM^{-M} (°)	Num (%)	GM (°)	GM^{-M} (°)
(a)	104	40.4	5.1	67.2	44.4	5.7	22.8
(b)	104	34.6	4.1	43.8	50.0	5.7	46.2
(c)	103	36.4	3.4	25.3	48.5	7.0	64.7
(d)	97	17.2	2.6	18.5	72.0	21.2	71.5
(e)	104	55.0	23.4	44.4	32.0	7.0	45.6
(f)	155	31.3	18.2	44.8	58.0	16.4	45.2

increasing section of the frontier is approximately equal to the angle between the CCR frontier and the vertical line.

- For cases 3a–3c, the GM values are less than 7° in both sections of the frontier, which shows that TRTS of these cases is constant.
- For case 3d, the GM value represents a small deviation from CTRS assumption in the increasing section of the frontier. The GM values for cases 3(e) and 3(f) show clearly that they are ITRS and VTRS, respectively.
- By using the GM, maximum GM values and percentage of DMUs, TRTS of all cases are displayed in the second and the third columns of Figure 3. The first column shows the original data, the second and third columns display the cases' TRTS in the increasing and decreasing sections of the frontier, respectively.

4.2 The real dataset; Maskan bank

To study the results of the proposed method, the real dataset of Maskan bank incorporated in 1979 is used. Its basic mission as a specialized bank is to support the development of housing and construction activities of the government and private sectors in Iran. Its total employees is about 10,860. The dataset includes the information of 1213 branches, which are subdivided into 37 management codes. The number of branches in each management code ranges from 9 to 85. For each branch, there are two inputs: personnel costs and location index and, three outputs: resources, expenses, and services. The personnel costs are normalized to 1,000,000. TRTS of this dataset is subjectively identified to be increasing.

Applying the angles method and the maximum angles method for each dataset requires that the number of DMUs to be at least equal to 3×(number of inputs+number of outputs). So, we applied the angles method and the maximum angles method for the management codes with at least 15 branches. For each management code and all the branches –which is represented by "All" in the first row–, the GM values (GM), the percentage of DMUs (Num), and the maximum GM values (GM^{-M}) are obtained in both sections of the frontier represented in Table 3.

Regarding Table 3, we mention some points:

- For all the subdivisions except codes 228 and 256, the percentage of DMUs in the decreasing section of the frontier is less than 10. Therefore, TRTS of all of the divisions except these two codes, are nondecreasing.
- The percentage of DMUs in the increasing section of the frontier for all the divisions ranges from 16.7 to 69. Therefore, the angles in the increasing section of the frontier are not negligible.

- For management codes 225, 242, and 245, the number of DMUs in the decreasing section is negligible. Moreover, their angles in the increasing section of the frontier are absolutely less than 10°. Thus, their TRTS are constant.

- The angles in the increasing section of the frontier for management codes 219, 224, and 249 are a bit less than 10°. So, the increasing feature of their TRTS is weak.

- Regarding the percentage of DMUs and the angles in the increasing and decreasing sections of the frontier for management codes 228 and 256, these two codes are VTRS.

- For all the divisions except the management codes 219, 224, 225, 228, 242, 245, 249, and 256, the angles in the increasing section of the frontier, range from 10.0° to 22.05°, which represent that their TRTS are increasing.

- TRTS of the entire branches; that is, "All", and most of the management codes are increasing.

- By using the GM and maximum GM values, we can display TRTS of each management code. Here, TRTS of "All" and management code 211 are displayed in Figure 4.

- From Figure 4, it can be seen that for both cases, there is a large angle from CTRS frontier in the decreasing section of the frontier. However, the narrow line shows that this large angle is inconsiderable.

5 Conclusion

Recently, the angles method, which is a heuristic method based on data mining, has been proposed for identifying a dataset's TRTS. In this paper, we extended the angles method by proposing the maximum angles method to measure the maximum deviation from CTRS assumption. We use the angles and the maximum angles for heuristically visualizing TRTS in a two-dimensional space. The methods are validated by 6 one input/one output cases. Also, these methods are applied for the dataset of Maskan bank of Iran. The results showed that all bank branches and the most of the management codes are ITRS. Moreover, the dataset of two cases are depicted in a two-dimensional space by the obtained results.

Table 3: The results of the angles method and the maximum angles method for Maskan bank dataset

		Increasing section			Decreasing section		
Code	DMUs	Num (%)	GM (°)	GM^{-M} (°)	Num (%)	GM (°)	GM^{-M} (°)
All	1213	66.2	15.6	44.1	3.4	12.3	57.1
211	85	53.8	13.3	40.4	5.1	17.6	51.8
212	63	63.5	14.4	39.4	1.9	4.5	49.7
214	68	51.5	12.4	42	1.5	19.0	43.1
215	38	69.0	19.2	47.9	0.0	–	48.5
216	69	35.5	15.4	34.8	6.5	8.0	48.7
217	48	58.1	22.0	46.5	0.0	–	41.4
218	83	46.2	19.0	45.7	0.0	–	34.6
219	51	23.4	9.6	30.8	4.3	7.6	48.9
221	47	52.4	12.8	33.9	2.4	0.9	55.7
222	47	43.2	15.1	39.2	0.0	–	9.3
224	49	40.9	9.6	34.1	0.0	–	6.5
225	28	53.3	7.5	30	6.7	19.4	59.4
226	45	45.0	16.1	48.1	0.0	–	39.9
227	19	44.4	17.4	48.7	0.0	–	27.8
228	34	20.0	17.6	35.3	20.0	9.9	35.4
229	25	61.1	17.7	40.2	0.0	–	37.9
241	25	16.7	11.7	46.2	0.0	–	38.6
242	24	46.7	7.4	24.4	0.0	–	33.6
244	24	47.1	12.3	26.2	0.0	–	17.1
245	22	50.0	6.2	21.7	0.0	–	34.4
246	20	44.4	10.0	37	0.0	–	31.0
247	19	50.0	15.0	39.8	0.0	–	18.7
248	29	50.0	11.5	25.9	0.0	–	26.2
249	21	20.0	9.6	31.7	0.0	–	27.7
252	50	40.0	12.7	43.5	6.7	31.0	39.1
254	21	50.0	18.6	53.5	0.0	–	38.3
255	17	28.6	11.4	23.9	0.0	–	17.1
256	22	22.2	11.7	35.7	11.1	18.0	61.6
264	49	42.9	14.9	45.1	0.0	–	19.5

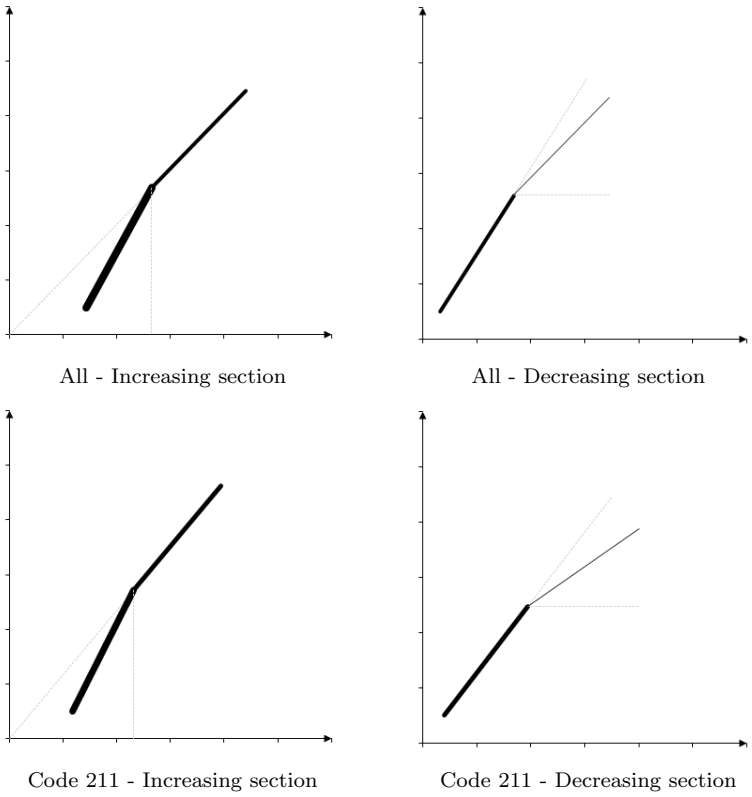


Figure 4: Displaying TRTS in the increasing and the decreasing sections of the frontier for all branches ("All") and management code 211. The horizontal and the vertical coordinates are the input and output, respectively.

References

1. Alirezaee, M., Hajinezhad, E., and Paradi, J. C. Objective identification of technological returns to scale for data envelopment analysis models. European Journal of Operational Research (2017).

2. Allahyar, M., and Rostamy-Malkhalifeh, M. Negative data in data envelopment analysis: Efficiency analysis and estimating returns to scale. Computers & Industrial Engineering 82 (2015), 78–81.

3. Banker, R. D. Estimating most productive scale size using data envelopment analysis. european journal of Operation Researchs 17 (1984), 35–44.

4. Banker, R. D. Hypothesis tests using data envelopment analysis. Journal of Productivity Analysis 7, 2 (1996), 139–159.

5. Banker, R. D., Charnes, A., and Cooper, W. W. Some models for estimating technical and scale ineffciencies in data envelopment analysis. Management Science 30, 9 (1984).

6. Banker, R. D., Cooper, W. W., Seiford, L. M., Thrall, R. M., and Zhu, J. Returns to scale in different dea models. European Journal of Operational Research 154, 2 (2004), 345–362.

7. Banker, R. D. Maximum likelihood, consistency and data envelopment analysis. a statistical foundation. Management Science 39, 10 (1993), 1265–1273.

8. Banker, R. D., and Natarajan, R. Statistical tests based on DEA efficiency scores. Springer, 2011, pp. 273–295.

9. Banker, R. D., and Thrall, R. M. Estimation of returns to scale using data envelopment analysis. European Journal of Operational Research 62, 1 (1992), 74–84.

10. Bougnol, M. L., and Dul, J. H. Anchor points in dea. European Journal of Operational Research 192, 2 (2009), 668–676.

11. Charnes, A., Cooper, W. W., and Rhodes, E. Measuring the efficient of decision making unit. European Journal of Operation Research 2 (1978), 429–444.

12. Divine, J. D. Efciency Analysis and Management of Not for Prot and Governmentally Regulated Organizations. Thesis, 1986.

13. Fre, R., Grosskopf, S., and Lovell, C. The measurement of efciency of production. Boston: Kluwer Nijhoff Publishing, 1985.

14. Fre, R., Grosskopf, S., and Lovell, C. A. K. Production Frontiers. Cambridge University Press, 1994.

15. Kerstens, K., and Vanden Eeckaut, P. Estimating returns to scale using non-parametric deterministic technologies: A new method based on goodness-of-fit. European Journal of Operational Research 113, 1 (1999), 206–214.

16. Korhonen, P. J., Soleimani-Damaneh, M., and Wallenius, J. On ratio-based rts determination: An extension. European Journal of Operational Research 231, 1 (2013), 242–243.

17. Krivonozhko, V. E., Frsund, F. R., and Lychev, A. V. Measurement of returns to scale using non-radial dea models. European Journal of Operational Research 232, 3 (2014), 664–670.

18. Lotfi, F. H., Jahanshahloo, G. R., and Esmaeili, M. An alternative approach in the estimation of returns to scale under weight restrictions. Applied mathematics and computation 189, 1 (2007), 719–724.

19. Podinovski, V. On the linearisation of reference technologies for testing returns to scale in fdh models. European Journal of Operational Research 152, 3 (2004), 800–802.

20. Simar, L., and Wilson, P. W. Non-parametric tests of returns to scale. European Journal of Operational Research 139, 1 (2002), 115–132.

21. Soleimani-damaneh, M., and Reshadi, M. A polynomial-time algorithm to estimate returns to scale in fdh models. Computers & Operations Research 34, 7 (2007), 2168–2176.

22. Sueyoshi, T., and Goto, M. Measurement of returns to scale and damages to scale for dea-based operational and environmental assessment: How to manage desirable (good) and undesirable (bad) outputs? European Journal of Operational Research 211, 1 (2011), 76–89.

23. Sueyoshi, T., and Goto, M. Returns to scale vs. damages to scale in data envelopment analysis: An impact of u.s. clean air act on coal-fired power plants. Omega 41, 2 (2013), 164–175.

24. Thanassoulis, E. Setting achievements targets for school children. Education Economics 7, 2 (1999), 101–119.

25. Tone, K. On returns to scale under weight restrictions in data envelopment analysis. Journal of Productivity Analysis 16, 1 (2001), 31–47.

26. Wu, J., and An, Q. Slacks-based measurement models for estimating returns to scale. International Journal of Information and Decision Sciences 5, 1 (2013), 25–35.

Iranian Journal of Numerical Analysis and Optimization
Vol. 8, No. 2, (2018), pp 75–93
DOI:10.22067/ijnao.v8i2.49973

Explicit and implicit schemes for fractional–order Hantavirus model

Mevlüde Yakıt Ongun* and Damla Arslan

Abstract

In this paper, the fractional–order form of a mouse population model is introduced. Some explicit and implicit schemes, which are Theta methods and nonstandard finite difference (NSFD) schemes, are implemented to give a numerical solution of nonlinear ordinary differential equation system named Hantavirus epidemic model. These methods are compared and discussed that the method preserves the positivity properties of the integer order system. The numerical solutions are illustrated by means of some graphs. Numerical results of explicit and implicit methods denote that these methods are easy and accurate when applied to fractional–order Hantavirus model.

Keywords: Explicit and implicit methods; Theta method; Nonstandard finite difference scheme; Fractional–order nonlinear differential equation systems; Mouse population model.

1 Introduction

Although fractional calculus has a long history, its applications to natural science are just a recent focus of interest. The description of some phenomena is more accurate when the fractional derivative is used. In many scientific area specially in physics, chemistry, and engineering, the fractional differential equations (FDEs) become a popular subject, which are increasingly used to model problems in a number of research areas including dynamical systems, mechanical systems, signal processing, electronic circuit theory, control theory, chaos synchronization, mechanics, seismology, and many others.

*Corresponding author
Received 19 September 2015; revised 18 April 2018; accepted 16 May 2018
Mevlüde Yakıt Ongun
Department of Mathematics, Suleyman Demirel University, Isparta, Turkey.
e-mail: mevludeyakit@sdu.edu.tr

Damla Arslan
Graduate School of Natural and Applied Sciences, Suleyman Demirel University, Isparta, Turkey.

Some of these studies may be found in [4] and [35]. Lots of books written by the authors Podlubny [30], Lubich [19], Miller and Ross [24], Oldham and Spainer [28], Diethelm et al. [8], Samko et al. [34] played a significant role to understand the subject of FDEs and gave some methods for solutions.

Several methods, which are the Adams–Moulton method [14], [12], the homotopy perturbation method [25], the Adomian decomposition method [25], [26], the variational iteration method [36], the differential transform method [3], [11], the operational method [20], the predictor corrector methods [7], [8], [13], the product integration rules [15], [38], nonstandard finite difference scheme [29] have been used to solve FDEs. Numerical solution methods are frequently referred to as being explicit or implicit. These situations are being encountered numerical solutions of FDEs and equation systems.

The first and important study, which is relevant with Hantavirus model, is given by [2]. The stability analysis of model, which is discretized with nonstandard finite difference scheme, is given by [5], [6]. Further, variational iteration method [16], exponential matrix method (EMM) [39], multistage differential transformation method [17] have been used to solve Hantavirus model and FDE solutions of this model studied by [31], [37].

The effect of the environmental parameter on the Hantavirus infection through a fractional–order SI model was studied by Rida et al.(2012). They presented a fractional–order model of the Hantavirus infection in terms of simple differential equations involving the mice population. They studied that the effect of changes in ecological conditions and diversity of habitats can be observed by varying the value of the environmental parameter k. They used a generalized Euler method (GEM) to obtain an analytic approximate solution of the model [37].

In this paper, new numerical methods determined for fractional–order nonlinear differential equation model known as Hantavirus epidemic model. These model based on a nonlinear differential equation of order p, where p is a constant in range $0 < p < 1$. Some explicit and implicit methods such as theta method and nonstandard finite difference schemes are studied for the numerical solution of the fractional–order model. Especially the purpose of the studying to these two methods that both methods are more general face of classical methods and both methods allow to some arbitrary choice.

The following model is given in [2] has been used in the study of Hantavirus epidemics:

$$\begin{aligned} \frac{dM_s}{dt} &= bM - cM_s - \frac{M_s M}{K} - aM_s M_I, \\ \frac{dM_I}{dt} &= -cM_I - \frac{M_I M}{K} + aM_s M_I, \end{aligned} \tag{1}$$

where

M_s : the population of susceptible mice ($M_s \geqslant 0$),

M_I : the population of infected mice ($M_I \geqslant 0$),

b : the birth rate,

c : the death rate,

K : the carrying capacity of the environment.

In this model, mice movement as a process of diffusion is ignored and whole population is composed of two cases of mice, susceptible and infected. For the systems (1), the total population $M = M_s + M_I$ satisfying the logistic differential equation

$$\frac{dM}{dt} = (b-c)M - \frac{M^2}{K}. \tag{2}$$

The carrying capacity for the total population is $M^* = (b-c)K$. The critical value of the carrying capacity is

$$K_c = \frac{b}{a(b-c)}.$$

Let us $M_s = x, M_I = y$ in equation (1),

$$\begin{aligned} \frac{dx}{dt} &= (b-c)x + by - \frac{x^2}{K} - \left(\frac{1+aK}{K}\right)xy, \\ \frac{dy}{dt} &= -cy - \frac{y^2}{K} - \left(\frac{1-aK}{K}\right)xy. \end{aligned} \tag{3}$$

The dynamics of the continuous system (3) have been given in [2,5,6] as the following:

i) If $b \leq c$, then the system has a unique equilibrium $(0,0)$, and it is globally asymptotically stable.

ii) If $b > c$ and $K \leq \frac{b}{a(b-c)}$, then the system has two equilibria: $(0,0)$, which is unstable, and $(K(b-c),0)$, which is globally asymptotically stable.

iii) If $b > c$ and $K > \frac{b}{a(b-c)}$, then the system has three equilibria: $(0,0)$ and $(K(b-c),0)$, which are unstable; and $\left(\frac{b}{a}, K(b-c) - \frac{b}{a}\right)$, which is globally asymptotically stable.

In this article,the fractional–order Hantavirus model examined, which given as below:

$$\begin{aligned} {}^{C}D^{p}_{t_0}x(t) &= (b-c)x(t) + by(t) - \frac{x(t)^2}{K} - \left(\frac{1+aK}{K}\right)x(t)y(t), \\ {}^{C}D^{p}_{t_0}y(t) &= -cy(t) - \frac{y(t)^2}{K} - \left(\frac{1-aK}{K}\right)x(t)y(t), \\ x(0) = x_0, &\quad y(0) = y_0, \end{aligned}$$

where noninteger derivative is defined as the Caputo derivative. The fractional–order equation system is introduced and denoted that this model has a unique solution with given initial conditions while $t \geq 0$. Some explicit and implicit schemes are implemented to study the dynamics behaviors of fractional order Hantavirus epidemic system. Some numerical solutions illustrated by means of some graphs are provided in next sections.

2 Fractional derivatives

On this section, some definitions and relationship of fractional order integration and fractional–order differentiation mentioned [29].

The Grünwald–Letnikov (GL) operator of order $p > 0$ is defined as

$$ {}_{GL}D^{p}_{t_0} f(t) = \lim_{N\to\infty} h_N^{-p} \sum_{j=0}^{N} w_j^{(p)} f(t - jh_N) $$

with

$$ w_j^{(p)} = (-1)^j \binom{p}{j} = \frac{\Gamma(j-p)}{\Gamma(-p)\Gamma(j+1)}. $$

Note that the weights $w_j^{(p)}$ are the coefficient in the power series expansion of $(1-\xi)^p$; that is,

$$ (1-\xi)^p = \sum_{j=0}^{\infty} w_j^{(p)} \xi^j $$

and they can be evaluated recursively by means of the recurrence

$$ w_0^{(p)} = 1, \quad w_j^{(p)} = \left(1 - \frac{1+p}{j}\right) w_{j-1}^{(p)}, \quad j = 1, 2, \ldots. \tag{4} $$

The relationship between Riemann–Liouville (RL) and Caputo definitions is

$$ {}_{C}D^{p}_{t_0} f(t) = {}_{RL}\ D^{p}_{t_0} \left(f(t) - T_{m-1}[f; t_0] \right), $$

where $T_{m-1}[f; t_0]$ is the $(m-1)$th degree Taylor polynomial for f centered at t_0

$$ T_{m-1}[f; t_0](t) = \sum_{k=0}^{m-1} \frac{t^k}{k!} f^{(k)}(t_0). $$

While, $0 < p < 1$, it is $m = 1$ and $T_0[f; t_0](t) = f(t_0)$. Thus, we obtained

$$ {}_{C}D^{p}_{t_0} f(t) = {}_{RL}\ D^{p}_{t_0} (f(t) - f(t_0)). $$

Under suitable assumptions of regularity, the RL and GL operators coincide; that is,

$$ {}_{RL}D^{p}_{t_0} f(t) =_{GL} D^{p}_{t_0} f(t). $$

A consequence of the above two points is that in general

$$ {}_{C}D^{p}_{t_0} f(t) =_{GL} D^{p}_{t_0} f(t) \left(f(t) - T_{m-1}[f; t_0] \right), $$

and while $0 < p < 1$ it is

$$ {}_{C}D^{p}_{t_0} f(t) =_{GL} D^{p}_{0,t}(f(t) - f(t_0)). \tag{5} $$

In case it used in differential equations of fractional order, the Caputo definition is preferable since it allows the couple the equation with initial conditions of classical type (i.e., initial condition of Cauchy type). In this way we can obtain

$$ \begin{cases} {}_{C}D^{p}_{t_0} f(t) = f(t, y(t)), \\ y(t_0) = y_0. \end{cases} $$

In case we use the RL definition; we can couple different initial conditions, as

$$ \begin{cases} {}_{RL}D^{p}_{t_0} f(t) = f(t, y(t)), \\ \lim\limits_{t \to t_0^+} J^{1-p}_{t_0} y(t) = b_1, \end{cases} $$

which does not have a clear physical meaning. Therefore, they are not useful for practical applications.

Lemma 1. *Let $0 < p < 1$, and let $w_n^{(p)}$ be the coefficients in the power series expansion of $(1 - \xi)^{\alpha}$ given in $w_j^{(p)}$. Then for any $n = 1, 2, \ldots,$*

a) $-1 < w_1^{(p)} < 0,$

b) $0 < w_1^{(p-1)} < 1.$

Proof. The proof is an immediate consequence of the recursive relationship stated in (4). □

3 Fractional order Hantavirus epidemics

In this section, the fractional order applied in the model of Hantavirus epidemics [2]. The new system is given as the set of fractional differential equations (FDEs):

$$
\begin{aligned}
{}^{C}D^{p}_{t_0}x(t) &= (b-c)x(t) + by(t) - \frac{x(t)^2}{K} - \left(\frac{1+aK}{K}\right)x(t)y(t),\\
{}^{C}D^{p}_{t_0}y(t) &= -cy(t) - \frac{y(t)^2}{K} - \left(\frac{1-aK}{K}\right)x(t)y(t),
\end{aligned} \tag{6}
$$

$$
x(0) = x_0, \quad y(0) = y_0, \tag{7}
$$

where ${}_{C}D^{p}_{0,t}$ denotes the fractional derivative operator, with respect to the origin, according to Caputo's definition [30] and $x(t)$ and $y(t)$ are activator and inhibitor variables. The fractional derivatives are used to describe nonhomogeneous character of the ecosystems, with respect to the presence of competitors. The parameter p denotes the density of competitor species in the system. The reason, to use fractional order differential equations (FODE), is to be naturally related with systems with memory, which exists in most biological systems. Also FODE are closely related to fractals, which are abundant in biological systems [17]. The results derived of the fractional system (3) and (4) are more general nature.

When the power exponent is $p = 1$, this corresponds to equation (3) and varies competitor's populations when $0 < p < 1$. While $p > 1$, the density of competitor or alien species will increase in the populations [1]. The stability analysis given by the study of [29].

For non–negative solutions, denote $\mathbb{R}^2_+ = \left\{x \in \mathbb{R}^2 \mid x \geq 0\right\}$, and let $x(t) = (X, Y)^T$. To prove the main theorem, we need following lemma and corollary given by [9, 27].

Lemma 2 (Generalized Mean Value Theorem)**.** *Suppose that $f(x) \in C[a, b]$ and $D^p_a f(x) \in C[a, b]$ for $0 < p \leq 1$; then we have*

$$
f(x) = f(a) + \frac{1}{\Gamma(p)}\left(D^p_a f\right)(\xi)(x-a)^p
$$

with $a \leq \xi \leq x$ for all $x \in (a, b]$.

Corollary 1. *Suppose that $f(x) \in C[a, b]$ and $D^p_a f(x) \in C[a, b]$ for $0 < p \leq 1$. If $D^p_a f(x) \geq 0$, for all $x \in (a, b)$, then $f(x)$ is nondecreasing for each $x \in [a, b]$. If $D^p_a f(x) \leq 0$, for all $x \in (a, b)$, then $f(x)$ is nonincreasing for each $x \in [a, b]$.*

Theorem 1. *There is a unique solution $x(t) = (X, Y)^T$ to equation (6) with initial condition (7) on $t \geq 0$, and the solution will remain in $\mathbb{R}^2_+$. Furthermore, X and Y are all bounded by K_c.*

Proof. We know that the solution on $(0, \infty)$ solving the initial value problem (6)–(7) is not only existence but also unique [18].

From equation (6), we find

$$D^p X \mid_{X=0} = bY \geqslant 0, D^p Y \mid_{Y=0} = 0.$$

By Corollary 1, the solution will remain in $\mathbb{R}^2_+$. From [2], X and Y are bounded by K_c. □

To evaluate the equilibrium points, let $D^p x = 0$ and $D^p y = 0$. Then $E_0 = (0, 0)$, $E_1 = (K(b-c), 0)$, and$E_2 = (\frac{b}{a}, K(b-c) - \frac{b}{a})$ are the equilibrium points. To give more detail about the local behavior near the equilibria, we find the Jacobian matrix of equation (6) at each equilibrium point:

$$J(E_0) = \begin{pmatrix} b-c & b \\ 0 & -c \end{pmatrix}, J(E_1) = \begin{pmatrix} -(b-c) & c - aK(b-c) \\ 0 & -b + aK(b-c) \end{pmatrix}$$

and

$$J(E_2) = \begin{pmatrix} -aK(b-c) + b(1 - \frac{1}{aK}) & -\frac{b}{aK} \\ aK(b-c) + \frac{b}{aK} - 2b + c & -b(1 - \frac{1}{aK}) + c \end{pmatrix}.$$

The equilibrium point E_0 is locally asymptotically stable, if all of the eigenvalues λ_i $(i = 1, 2)$ of the Jacobian matrix $J(E_i)$ for $i = 0, 1, 2$ satisfy the following condition [10, 21]:

$$|\arg(\lambda_i)| > \frac{p\pi}{2}. \tag{8}$$

The eigenvalues of $J(E_0)$ are $\lambda_1 = (b - c)$ and $\lambda_2 = -c$. If $b \geq c$, then $\arg(\lambda_1) = 0, \arg(\lambda_2) = \pi$, and E_0 is unstable. If $b < c, \arg(\lambda_1) = \arg(\lambda_2) = \pi$, and equation (8) is correct, then E_0 is asymptotically stable.

Now consider the case $b > c$ and the equilibrium point E_1. The eigenvalues of $J(E_1)$ are $\lambda_1 = -(b - c)$ and $\lambda_2 = -b + aK(b - c)$. $\arg(\lambda_1) = \pi$ and for $K < K_c$, $\arg(\lambda_2) = \pi$, thus E_1 is asymptotically stable. If $K > K_c$, then E_1 is unstable.

Finally, consider the case $b > c$ for the equilibrium point E_2. If $K > K_c$, then $\lambda_1 = -(b - c)$ and $\lambda_2 = b - aK(b - c)$ are negative eigenvalues and $\arg(\lambda_1) = \arg(\lambda_2) = \pi$; consequently, E_2 is asymptotically stable.

4 Theta method for fractional order models

In this section, theta method ($\theta - -method$) is described and studied for the numerical solution of fractional–order equation systems by the way of Hantavirus model. Theta method is known as the weighted method and general form given by

$$D^p y(t_n) = \theta f(t_n, y_n) + (1 - \theta) f(t_{n-1}, y_{n-1}) \tag{9}$$

while $\theta \in [0,1]$. There are two specific values of θ: while $\theta = 0$, we obtain the forward explicit Euler method and while $\theta = 1$ yields forward implicit Euler method. Note that $\theta = \frac{1}{2}$ in (9) corresponds the so–called mid–point rule and trapezoidal rule, respectively.

For numerical computation the Grünwald–Letnikov (GL) operator is truncated; we fixed a step–size $h > 0$ and $t_n = t_0 + hn, N = (T - t_0)/h$.

We put the left–hand–side with the forward discrete derivative with

$$h^{-p}\sum_{j=0}^{n} w_j^p (y_{n-j} - y_0) = \theta f(t_n, y_n) + (1-\theta) f(t_{n-1}, y_{n-1}).$$

Case 1. While $\theta = 0$ on (9), in case we use it for the model equation of (6), we can get

$$\begin{cases} h^{-p}\sum\limits_{j=0}^{n} w_j^p (x_{n-j} - x_0) = (b-c)x_{n-1} + by_{n-1} - \frac{x_{n-1}^2}{K} - \left(\frac{1+aK}{K}\right) x_{n-1}y_{n-1}, \\ h^{-p}\sum\limits_{j=0}^{n} w_j^p (y_{n-j} - y_0) = -cy_{n-1} - \frac{y_{n-1}^2}{K} - \left(\frac{1-aK}{K}\right) x_{n-1}y_{n-1}. \end{cases}$$

Here, the left–hand derivatives are Grünwald–Letnikov derivatives.

$$\begin{cases} \sum\limits_{j=0}^{n} w_j^p x_{n-j} - x_0 \sum\limits_{j=0}^{n} w_j^p \\ \qquad\qquad = h^p \left[(b-c)x_{n-1} + by_{n-1} - \frac{x_{n-1}^2}{K} - \left(\frac{1+aK}{K}\right) x_{n-1}y_{n-1}\right], \\ \sum\limits_{j=0}^{n} w_j^p y_{n-j} - y_0 \sum\limits_{j=0}^{n} w_j^p = h^p \left[-cy_{n-1} - \frac{y_{n-1}^2}{K} - \left(\frac{1-aK}{K}\right) x_{n-1}y_{n-1}\right]. \end{cases}$$

In case we leave x_n and y_n alone on the left side,

$$\begin{cases} x_n = h^p \left[(b-c)x_{n-1} + by_{n-1} - \frac{x_{n-1}^2}{K} - \left(\frac{1+aK}{K}\right) x_{n-1}y_{n-1}\right] \\ \qquad\qquad - \sum\limits_{j=1}^{n} w_j^{(p)} x_{n-j} + w_n^{(p-1)} x_0, \\ y_n = h^p \left[-cy_{n-1} - \frac{y_{n-1}^2}{K} - \left(\frac{1-aK}{K}\right) x_{n-1}y_{n-1}\right] - \sum\limits_{j=1}^{n} w_j^{(p)} y_{n-j} - w_n^{(p-1)} y_0. \end{cases} \tag{10}$$

Since $w_1^{(\alpha)} = -p$, we obtain

$$\widetilde{x}_{n-1} = w_1^{(p-1)} x_0, \quad n = 1 \quad \textit{and} \quad w_1^{(p-1)} x_0 - \sum_{j=2}^{n} w_j^{(p)} x_{n-j}, \quad n \geq 2,$$

and similarly we can say for $\widetilde{y}_{n-1}$. While

$$\begin{cases} w_n^{(p-1)}x_0 - X_n = \widetilde{x}_{n-1} + px_{n-1}, \\ w_n^{(p-1)}y_0 - Y_n = \widetilde{y}_{n-1} + py_{n-1}, \end{cases} \tag{11}$$

and

$$X_n = \sum_{j=1}^{n} w_j^{(p)} x_{n-j}, \qquad Y_n = \sum_{j=1}^{n} w_j^{(p)} y_{n-j} \ .$$

We get (11) in the equations system (10), and we obtain iteration equations as:

$$\begin{cases} x_n = \widetilde{x}_{n-1} + px_{n-1} \\ \qquad\qquad +h^p \left[(b-c)x_{n-1} + by_{n-1} - \frac{x_{n-1}^2}{K} - \left(\frac{1+aK}{K}\right) x_{n-1}y_{n-1}\right], \\ y_n = \widetilde{y}_{n-1} + py_{n-1} + h^p \left[-cy_{n-1} - \frac{y_{n-1}^2}{K} - \left(\frac{1-aK}{K}\right) x_{n-1}y_{n-1}\right]. \end{cases} \tag{12}$$

Case 2. While $\theta = 1$ on (9), use it for Equation (6), we obtain implicit form

$$\begin{cases} h^{-p} \sum\limits_{j=0}^{n} w_j^p (x_{n-j} - x_0) = (b-c)x_n + by_n - \frac{x_n^2}{K} - \left(\frac{1+aK}{K}\right) x_n y_n, \\ h^{-p} \sum\limits_{j=0}^{n} w_j^p (y_{n-j} - y_0) = -cy_n - \frac{y_n^2}{K} - \left(\frac{1-aK}{K}\right) x_n y_n. \end{cases}$$

In case we leave x_n and y_n alone on the left side,

$$\begin{cases} x_n - h^p \left[(b-c)x_n + by_n - \frac{x_n^2}{K} - \left(\frac{1+aK}{K}\right) x_n y_n\right] + X_n - w_n^{(p-1)}x_0 = 0, \\ y_n - h^p \left[-cy_n - \frac{y_n^2}{K} - \left(\frac{1-aK}{K}\right) x_n y_n\right] + Y_n - w_n^{(p-1)}y_0 = 0. \end{cases}$$

While system has implicit form, Newton–Raphson method needed to solve implicit system by converting to explicit form. Here, the implicit form, x_n and y_n connected to the explicit form of the first equation to be able to resort to Newton–Raphson method:

$$\begin{cases} f(z) = z + X_n - w_n^{(p-1)}x_0 - h^p \left[(b-c)z + bu - \frac{z^2}{K} - \left(\frac{1+aK}{K}\right) zu\right], \\ g(u) = u + Y_n - w_n^{(p-1)}y_0 - h^p \left[-cu - \frac{u^2}{K} - \left(\frac{1-aK}{K}\right) zu\right]. \end{cases}$$

Regarding to Newton –Raphson method, iterative equations as:

$$z_{i+1} = z_i - \frac{f(z_i)}{f'(z_i)}, \qquad i = 0, 1, \ldots, n,$$
$$u_{i+1} = u_i - \frac{g(u_i)}{g'(u_i)}, \qquad i = 0, 1, \ldots, n.$$

Here, we can write initial estimations as $z_0 = x_{n-1}$ and $u_0 = y_{n-1}$. Thus we obtain

$$\begin{cases} z_{i+1} = z_i - \dfrac{z_i - \widetilde{x}_{n-1} - pz_0 - h^p\left[(b-c)z_i + bu_i - \frac{z_i^2}{K} - \left(\frac{1+aK}{K}\right)z_i u_i\right]}{1 - h^p\left[(b-c) - \frac{2z_i}{K} - \left(\frac{1+aK}{K}\right)u_i\right]}, \\ u_{i+1} = u_i - \dfrac{u_i - \widetilde{y}_{n-1} - pu_0 - h^p\left[-cu_i - \frac{u_i^2}{K} - \left(\frac{1-aK}{K}\right)z_i u_i\right]}{1 - h^p\left[-c - \frac{2u_i}{K} - \left(\frac{1-aK}{K}\right)z_i\right]}. \end{cases} \tag{13}$$

5 Nonstandard difference schemes methods for fractional order Hanta model

In this section, we proposed and discussed some NSFD schemes to discretize the fractional order nonlinear system (6). NSFD schemes applied in combination with the truncation of the GL operator. NSFD schemes were firstly proposed by Mickens for Ordinary Differential Equations and successively studied, for instance, in [22, 23]. The stability analysis of model which is discretized with nonstandard finite difference scheme is given by [5, 6].

Case 1: As a first nonstandard scheme, for discretization we make the replacement of the nonlinear term in the right–hand side of (6) by means of $x(t) \to x(t_{n-1})$, $y(t) \to y(t_{n-1})$, $x^2(t) \to x(t_n)x(t_{n-1})$, $x(t)y(t) \to x(t_n)y(t_{n-1})$.

Since $w_1^p = 1$, the discretized model is

$$x_n + X_n - w_n^{p-1}x_0 = \phi(h)\overset{*}{X},$$
$$y_n + Y_n - w_n^{p-1}y_0 = \phi(h)\overset{*}{Y},$$

which seem as explicit form and $\phi(h)$ is a denominator function and could be selected in a different way. Here, $X_n, Y_n, \overset{*}{X}$ and $\overset{*}{Y}$ are defined as:

$$X_n = \sum_{j=1}^{n} w_j^{(p)} x_{n-j}, \qquad Y_n = \sum_{j=1}^{n} w_j^{(p)} y_{n-j},$$

$$\begin{cases} \overset{*}{X} = (b-c)x_{n-1} + by_{n-1} - \frac{(x_{n-1}+y_{n-1})x_n}{K} - ax_n y_{n-1}, \\ \overset{*}{Y} = -cy_{n-1} - \frac{(x_{n-1}+y_{n-1})y_n}{K} - ax_n y_{n-1}, \end{cases}$$

$$\begin{cases} x_n = \frac{K\left(\phi(h)(b-c)x_{n-1}+\phi(h)by_{n-1}-X_n+w_n^{p-1}x_0\right)}{K+\phi(h)(x_{n-1}+y_{n-1})+K\phi(h)ay_{n-1}}, \\ y_n = \frac{K\left(-\phi(h)cy_{n-1}-a\phi(h)x_n y_{n-1}-Y_n+w_n^{p-1}y_0\right)}{K+\phi(h)(x_{n-1}+y_{n-1})}. \end{cases}$$

The result of explicit nonstandard finite differences scheme is

$$\begin{cases} x_n = \frac{K(\widetilde{x}_{n-1}+(p-c\phi(h))x_{n-1}+b\phi(h)(x_{n-1}+y_{n-1}))}{K+\phi(h)(x_{n-1}+y_{n-1})+aK\phi(h)y_{n-1}}, \\ y_n = \frac{K(\widetilde{y}_{n-1}+(p-c\phi(h))y_{n-1}+a\phi(h)x_n y_{n-1})}{K+\phi(h)(x_{n-1}+y_{n-1})}, \end{cases} \tag{14}$$

where we selected $\phi(h) = \left(\frac{\exp(h(b-c))-1}{b-c}\right)^p$. In [32, 33], authors have chosen $(xy) \to x(t+h)y(t)$, $x^2 \to x(t)x(t+h)$ and $y^2 \to y(t)y(t+h)$ for discretization.

Proposition 1. *Let $a, b, c, K > 0$, $b \leq c$, and let step size $0 < h \leq (p/c)^{(1/p)}$ for the schemes have the iterations x_n, y_n given by the non–negative schemes for any $x_0 \geq 0$ and $y_0 \geq 0$.*

Proof. We proceed by induction on n. □

Case 2: In the second nonstandard scheme we use the replacement $x(t) \to x(t_n)$, $y(t) \to y(t_n)$, $x^2(t) \to x(t_n)x(t_n)$, $x(t)y(t) \to x(t_n)y(t_n)$. By operating in a similar way as in the previous case for system (6) , we can see that is implicit form:

$$x_n + X_n - w_n^{p-1}x_0 = \phi(h)\overset{**}{X},$$
$$y_n + Y_n - w_n^{p-1}y_0 = \phi(h)\overset{**}{Y},$$

$$\begin{cases} \overset{**}{X} = (b-c)x_n + by_n - \frac{(x_n+y_n)x_n}{K} - ax_n y_n, \\ \overset{**}{Y} = -cy_n - \frac{(x_n+y_n)y_n}{K} - ax_n y_n. \end{cases}$$

If we leave alone to x_n on the left side and y_n on the right side, we obtain implicit form:

$$\begin{cases} x_n - \phi(h)\overset{**}{X} + X_n - w_n^{(p-1)}x_0 = 0, \\ y_n - \phi(h)\overset{**}{Y} + Y_n - w_n^{(p-1)}y_0 = 0. \end{cases}$$

While Newton–Raphson method is needed, here, the implicit form, x_n and y_n connected to the explicit form of the first equation to be able to resort to the method of Newton–Raphson, we leave alone on the left side:

$$\begin{cases} f(z) = z + X_n - w_n^{(p-1)} y_0 - \phi(h)\left[(b-c)z + bu - \frac{z^2}{K} - \left(\frac{1+aK}{K}\right) zu\right], \\ g(u) = u + Y_n - w_n^{(p-1)} u_0 - \phi(h)\left[-cu - \frac{u^2}{K} - \left(\frac{1-aK}{K}\right) zu\right]. \end{cases}$$

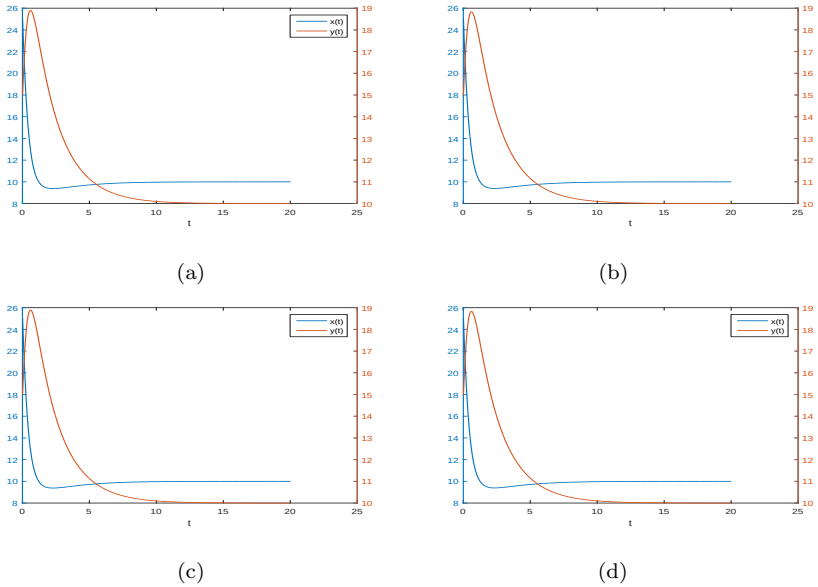
Regarding to Newton–Raphson method, the initial estimations as $z_0 = x_{n-1}$ and $u_0 = y_{n-1}$. In this way, we can obtain

$$\begin{cases} z_{i+1} = z_i - \frac{z_i - \widetilde{x}_{n-1} - pz_0 - \phi(h)\left[(b-c)z_i + bu_i - \frac{z_i^2}{K} - \left(\frac{1+aK}{K}\right) z_i u_i\right]}{1 - \phi(h)\left[(b-c) - \frac{2z_i}{K} - \left(\frac{1+aK}{K}\right) u_i\right]}, \\ u_{i+1} = u_i - \frac{u_i - \widetilde{y}_{n-1} - pu_0 - \phi(h)\left[-cu_i - \frac{u_i^2}{K} - \left(\frac{1-aK}{K}\right) z_i u_i\right]}{1 - \phi(h)\left[-c - \frac{2u_i}{K} - \left(\frac{1-aK}{K}\right) z_i\right]}. \end{cases} \tag{15}$$

6 Numerical simulations

In this section, numerical simulations of fractional–order Hantavirus model are presented by the solution of Theta method and NSFD schemes that are explicit and implicit form. We choose $a = 0.1$, $b = 1$, and $c = 0.5$ as used by study of [2]. The initial values is considered as $(x_0, y_0) = (25, 15)$; we fixed a step–size $h = 0.01$ and $N = 2000$. Various denominator functions listed in [29], we took into account the $\phi(h) = \left(\frac{\exp(h(b-c))-1}{b-c}\right)^p$. We choose $p = 1$ in Figure 1, $p = 0.6$ in Figure 2, $p = 0.3$ in Figure 3, $p = 1$ in Figure 4, and $(x_0, y_0) = (10, 10)$ and $K = 20$ used in Figure 5. In Figures 1–4, the top graphs in order of left–to–right are theta explicit and theta implicit methods, the bottom graphs in order of left–to–right are NSFD explicit and NSFD implicit methods. Figure 5 shows that for $p = 1$, $a = 0.1$, $b = 1$, $c = 0.5$ for x(t), relative errors between numerical schemes and exponential matrix method in [39]. All figures denoted the results of the simulation based on equation systems (12), (13), (14), and (15) the values are used as mentioned above. Equation systems (12), (13), (14), and (15) correspond to equations in (1).

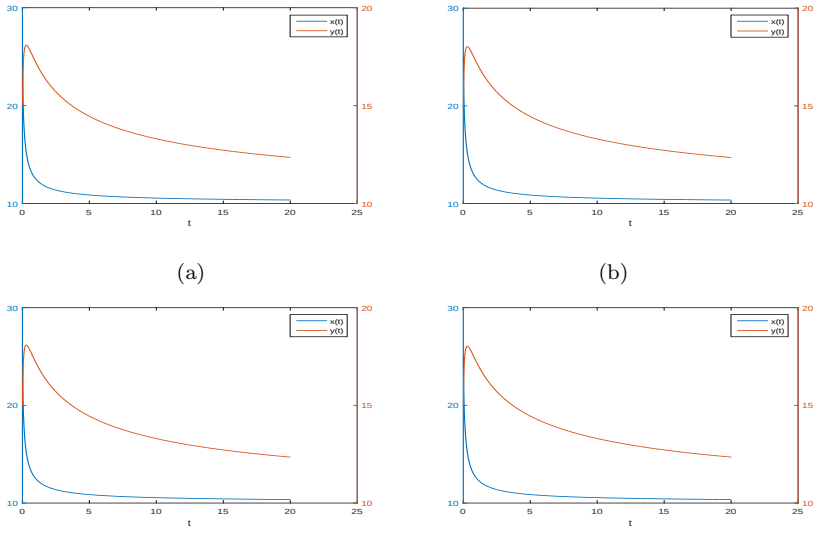
We compared CPU times of numerical methods on the Table 1. Here, we used $N = 1000$ for all methods except Runge Kutta methods that we used $N = 50$. As we have seen in the Table 1, we can say, in case each explicit and implicit methods evaluated between oneself has not extremely difference. However we cannot say the same for the Runge Kutta. On the Table 2, we denoted qualitative results of the fixed point E_2 for different time step sizes for $N = 1000$, $p = 1$, $(x_0, y_0) = (25, 15)$, and $K = 40$.



(a) (b)

(c) (d)

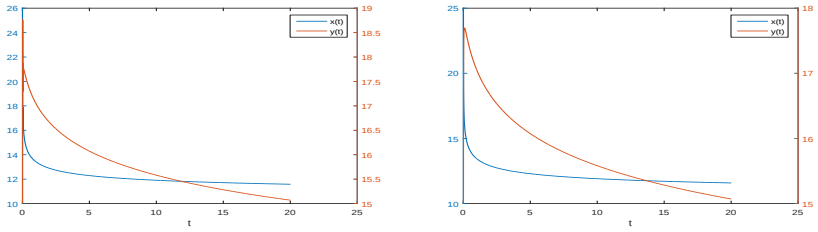
Figure 1: For p=1, Theta and NSFD Schemes in explicit and implicit form:
(a) Theta–explicit (b) Theta–implicit (c) NSFD–explicit (d) NSFD–implicit



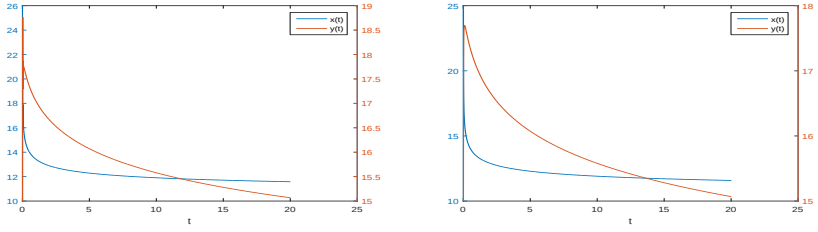
(a) (b)

(c) (d)

Figure 2: For p=0.6, Theta and NSFD Schemes in explicit and implicit form:
(a) Theta–explicit (b) Theta–implicit (c) NSFD–explicit (d) NSFD–implicit

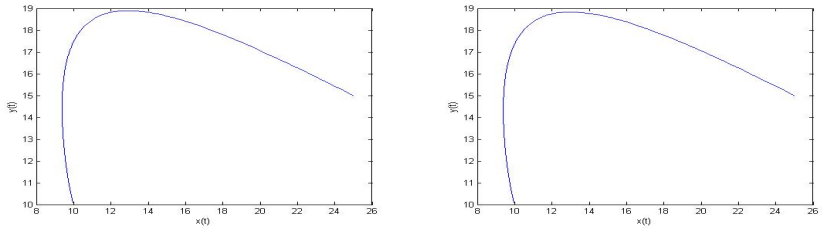


(a) (b)

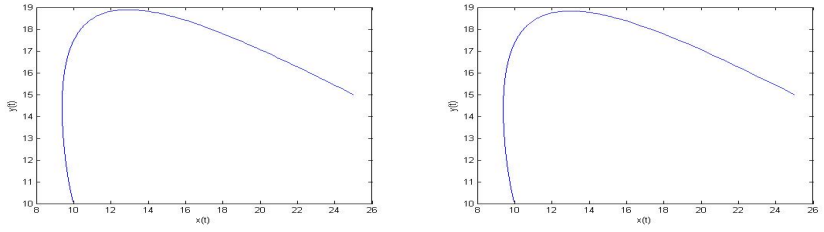


(c) (d)

Figure 3: For p=0.3, Theta and NSFD Schemes in explicit and implicit form:
(a) Theta–explicit (b) Theta–implicit (c) NSFD–explicit (d) NSFD–implicit



(a) (b)



(c) (d)

Figure 4: Phase–portraits for $p = 1$, $N = 2000$, $a = 0.1$, $b = 1$, $c = 0.5$, $h = 0.01$, $(x_0, y_0) = (25, 15)$, $K = 40$
(a) Theta–explicit (b) Theta–implicit (c) NSFD–explicit (d) NSFD–implicit

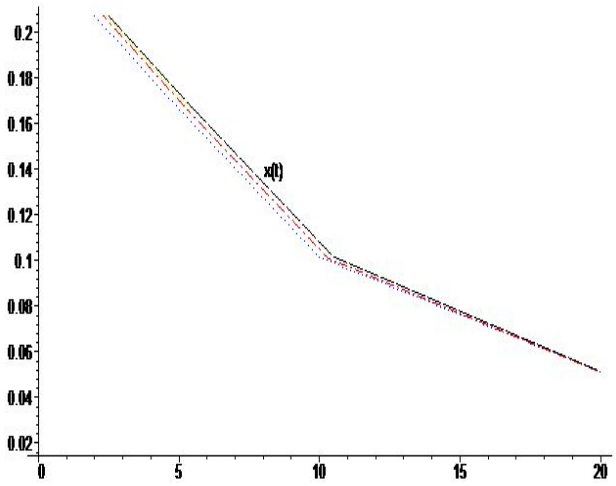


Figure 5: For $p = 1$, $a = 0.1$, $b = 1$, $c = 0.5$, $(x_0, y_0) = (10, 10)$, $K = 20$, for x(t), relative errors between numerical schemes and exponential matrix method in [39]
− − −− EMM and Theta-explicit
. . . . . EMM and NSFD implicit
..... EMM and Theta-implicit
_._._._._. EMM and NSFD-explicit

Table 1: CPU times (seconds) for $(x_0, y_0) = (25, 15), h = 0.01$

p	Theta Explicit	Theta Implicit	NSFD Explicit	NSFD Implicit	Runge Kutta
0.3	1.912602	2.216342	2.259337	1.813677	3.597516
0.6	1.909790	2.522203	2.331288	1.77084	3.6738
1	1.86862	2.273820	4.053474	1.767188	4.222919

Table 2: Qualitative results of the fixed point E_2 for different time step sizes for $N = 1000, P = 1, (x_0, y_0) = (25, 15), K = 40$(Con. and Div., resp., Convergence and Divergence)

h	Theta Explicit	Theta Implicit	NSFD Explicit	NSFD Implicit	Runge Kutta
0.01	*Con.*	*Con.*	*Con.*	*Con.*	*Con.*
0.1	*Con.*	*Con.*	*Con.*	*Con.*	*Con.*
1	*Div.*	*Con.*	*Div.*	*Con.*	*Div.*
10	*Div.*	*Con.*	*Div.*	*Con.*	*Div.*
100	*Div.*	*Con.*	*Div.*	*Con.*	*Div.*

7 Conclusion

In this paper, fractional–order form of the Hantavirus model is introduced. The stability of the equilibrium points is studied. Two numerical methods have been presented in the explicit and implicit form for solving fractional order Hantavirus epidemic model. When we use implicit solutions, we need to use the Newton–Raphson method to solve implicit system by converting to explicit form. For NSFD schemes, one denominator function and different nonlocal terms have been proposed, and the results have been compared with each other. As it seen clearly that explicit and implicit NSFD methods should be used ultimately depending on the choices of the nonlocal terms. On the other hand, explicit methods produce the same accuracy, but with less computational effort and time than implicit methods.

Acknowledgments

The authors thank to the referees for their valuable contributions to make this article more understandable and stronger.

References

1. Abdullah, F.A., Ismail, A.I.Md. *Simulations of the spread of the Hantavirus using fractional differential equations*, Matematika, 27 (2011), 149–158.

2. Abramson, G., Kenkre, V.M., *Spatio–temporal patterns in the Hantavirus infection*, Phys. Rev. E. 66 (2002), 011912.

3. Arikoglu, A., Ozkol, I., *Solution of fractional differential equation by using differential transforms method*, Chaos Solitons Fractals, 34 (2007), no. 5, 1473–1481.

4. Baleanu, D., Mohammadi, H., Rezapour, S., *Positive solutions of an initial value problem for nonlinear fractional differential equations*, Abstr. Appl. Anal.(2012), Art. ID 837437, 7.

5. Chen, M., Clemence, D.P., *Analysis of and numerical schemes for a mouse population model in Hantavirus Model*, J. Difference Equ. Appl.12 (2006), no. 9, 887–899.

6. Chen, M., Clemence, D.P., *Stability properties of a nonstandard finite difference scheme for a Hantavirus epidemic model*, J. Difference Equ. Appl. 12 (2006), no. 12, 1243–1256.

7. Diethelm, K., Freed, A. D., *The FracPECE subroutine for the numerical solution of differential equations of fractional order*, in Forschung und Wissenschaftliches Rechmen, 1998.

8. Diethelm, K., Ford, N.J., Freed, A.D., *A predictor–corrector approach for the numerical solution of fractional differential equations*. Nonlinear Dynam. 29 (2002), no. 1–4, 3–22.

9. Ding, Y., Ye, H., *A fractional–order differential equation model of HIV infection of CD4+T cells*, Math. Comput. Model. 50 (2009), 386–392.

10. El–Sayeda, A.M.A., El–Mesiry, A.E.M., El–Saka, H.A.A., *On the fractional order logistic equation*, Appl. Math. Lett. 20 (2007), 817–823.

11. Erturk, V.S., Momani, S., Odibat, Z., *Application of generalized differential transform method to multi–order fractional differential equations*, Commun. Nonlinear Sci. Numer. Simul. 13 (2008), no. 8, 1642–1654.

12. Garrappa, R., *On some explicit Adams multistep methods for fractional differential equations*, J. Comput. Appl. Math., 229 (2009), 392–399.

13. Garrappa, R., *On linear stability of predictor–corrector algorithms for fractional differential equations*, Int. J. Comput. Math. 87 (2010), no. 10, 2281–2290.

14. Garrappa, R., Galeone, L., *Fractional Adams–Moulton methods*, Math. Comput. Simulation,79 (2008), 1358–1367.

15. Garrappa, R., Popolizio, M., *On accurate product integration rules for linear fractional differential equations*, J. Comput. Appl. Math., 235 (2011) 1085–1097.

16. Goh, S. M., Ismail, A. I. M., Noorani, M. S. M., & Hashim, I., *Dynamics of the Hantavirus infection through variational iteration method*, Nonlinear Analysis: Real World Applications, 10 (2009), no. 4, 2171–2176.

17. Gökdoğan, A., Merdan, M., Yildirim, A., *A multistage differential transformation method for approximate solution of Hantavirus infection model*, Commun. Nonlinear Sci. Numer. Simul. 17 (2012), no 1, 1–8.

18. Lin, W., *Global existence theory and chaos control of fractional differential equations*, J. Math. Anal. Appl. 332 (2007), 709–726.

19. Lubich, C., *Discretized fractional calculus*, SIAM J. Math. Anal. 17 (1986), 704–719.

20. Luchko, Y., Gorenflo, R., *An operational method for solving fractional differential equations with the Caputo derivatives*, Acta Math. Vietnam 24 (1999), no. 2, 207–233.

21. Matignon, D., *Stability results for fractional differential equations with applications to control processing*, Computational Eng. in Sys. Appl., vol 2, Lille, France, 1996.

22. Mickens,R.E., *A nonstandard finite–difference scheme for the Lotka–Volterra systems*, Appl. Numer. Math., 45 (2003), 309–314.

23. Mickens, R.E., *Calculation of denominator functions for nonstandard finite difference schemes for differential equations satisfying a positivity condition*, Numer. Methods Partial Differ. Equ. 23(3) (2007) 672–691.

24. Miller, K.S., Ross, B., *An Introduction to the Fractional Calculus and Fractional Differential Equations*, John Willey & Sons, New York, 1993.

25. Momani, S., Odibat, Z., *Comparison between homotopy perturbation method and the variational iteration method for linear fractional partial differential equations*, Comput. Math. Appl., 54 (2007), no. 7–8, 910–919.

26. Momani, S., Odibat, Z., *Analytical approach to linear fractional partial differential equations arising in fluid mechanics*, Phys. Lett. A, 355 (2006), 271–279.

27. Obidat, Z.M., Shawagfeh, N.T., *Generalized Taylors formula*, Appl. Math. Comput. 186 (2007), 286–293.

28. Oldham, K.B., Spanier, J., *The Fractional Calculus, Mathematics in Science and Engineering*, Academic Press,New York, 1974.

29. Ongun, M.Y., Arslan, D., Garrappa,R. , *Nonstandard finite difference schemes for fractional order Brusselator system*, Adv. Difference Equ., doi: 10.1186/1687–1847 (2013), 102.

30. Podlubny, I., *Fractional Differential Equations*, Acad. Press, London, E2, 1999.

31. Rida, S. Z., El Radi, A. A., Arafa, A., Khalil, M. *The effect of the environmental parameter on the Hantavirus infection through a fractional–order SI model*, Int. J. Basic Appl. Sci. 1 (2012), no. 2, 88–99.

32. Roeger, L.-I., *Nonstandard finite difference schemes for the Lotka–Volterra systems: generalization of Mickens's method*, J. Difference Equ. Appl.12 (2006), no. 9, 937–948.

33. Roeger,L.-I., *Dynamically consistent discrete Lotka–Volterra competition models derived from nonstandard finite–difference schemes*, Discrete Cont. Dynam. Syst. Series B, 9 (2008), no. 2, 415–429.

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

35. Tenreiro Machado, J., Stefanescu, P., Tintareanu, O., Baleanu, D., *Fractional calculus analysis of cosmic microwave backgrounds*, Rom. Rep. Phys. 65 (2013), no. 1, 316–323.

36. Ünlü,C., Jafari, H., Baleanu, D., *Revised variational iteration method for solving systems of nonlinear fractional order differential equations*, Abstr. Appl. Anal., Article ID 461837, 7 pages, (2013).

37. Yaro, D., Omari–Sasu, S. K., Harvim, P., Saviour, A. W., & Obeng, B. A.,*Generalized Euler method for modeling measles with fractional differential equations*. Int. J. Innovative Research and Development, 4 (2015), no. 4.

38. Young,A., *Approximate product–integration*, Proc. Roy. Soc. London Ser. A. 224,(1954). 561–573.

39. Yüzbaşi, Ş., Sezer, M., *An exponential matrix method for numerical solutions of Hantavirus infection model*, Appl. Appl. Math. 8 (2013), no. 1.

Iranian Journal of Numerical Analysis and Optimization
Vol. 8, No. 2, (2018), pp 95–117
DOI:10.22067/ijnao.v8i2.60778

Discrete collocation method for Volterra type weakly singular integral equations with logarithmic kernels

P. Mokhtary*

Abstract

An efficient discrete collocation method for solving Volterra type weakly singular integral equations with logarithmic kernels is investigated. One of features of these equations is that, in general the first derivative of solution behaves like as a logarithmic function, which is not continuous at the origin. In this paper, to make a compatible approximate solution with the exact ones, we introduce a new collocation approach, which applies the Müntz-logarithmic polynomials(Müntz polynomials with logarithmic terms) as basis functions. Moreover, since implementation of this technique leads to integrals with logarithmic singularities that are often difficult to solve numerically, we apply a suitable quadrature method that allows the exact evaluation of integrals of polynomials with logarithmic weights. To this end, we first remind the well-known Jacobi–Gauss quadrature and then extend it to integrals with logarithmic weights. Convergence analysis of the proposed scheme are presented, and some numerical results are illustrated to demonstrate the efficiency and accuracy of the proposed method.

Keywords: Discrete collocation method; Müntz-logarithmic polynomials; Quadrature method; Volterra type weakly singular integral equations with logarithmic kernels.

1 Introduction

In this paper, we develop an approximate approach to obtain the numerical solution of the following Volterra type weakly singular integral equation with logarithmic kernel

*Corresponding author
Received 5 December 2016; revised 16 May 2018; accepted 13 June 2018
P. Mokhtary
Department of Mathematics, Faculty of basic Sciences, Sahand University of Technology, Tabriz, Iran. e-mail: mokhtary.payam@gmail.com mokhtary@sut.ac.ir

$$y(x) = g(x) + \int_0^x \ln(x-t) K(x,t) y(t) dt, \quad x \in \Omega = [0,1], \ 0 \le t \le x \le 1, \ (1)$$

where the continuous functions $g(x)$ and $K(x,t)$ are given and $y(x)$ is the unknown solution. Such kinds of equations arise from solution of Dirichlet's problem for the Laplace equation in two dimensions in terms of a single-layer logarithmic potential [24], solution of the reduced wave equation in two dimensions using an integral equation with a kernel, which can be expressed as a Hankel function of order zero that this kernel, has also a logarithmic singularity [25], investigation of electrostatic and low frequency electromagnetic problems [15], methods of computing the conformal mapping of a given domain [28], solution of electromagnetic scattering problems([2], [16]), determination of propagation of acoustical and elastical waves([5], [6]), boundary value problems of plane elasticity theory for regions with a defect [13], problems of diffraction by thin screens [12] and so on.

Weakly singular integral equations with logarithmic kernels are usually difficult to solve analytically; so it is necessary to provide reliable numerical techniques. There are several approximate methods proposed to obtain the numerical solution of these types of equations, which we refer to some of them. In [4], authors designed a computational meshless discrete Galerkin method to solve the second kind Fredholm integral equations with logarithmic kernels. In [17], authors developed a collocation method for the numerical solution of a special integro-differential equation with logarithmic kernel using airfoil polynomials of the first kind. A collocation method based on the periodic splines was introduced in [11] to solve some logarithmic kernel integral equations on open arcs. In [29], authors studied a special integral equation with logarithmic kernel and solved it using product integration method. In [14], a piecewise Chebyshev expansion was considered to solve Volterra integral equations with logarithmic singularities in their kernels. The properties of the integro-differential equations of the convolution on a finite interval with kernel having a logarithmic singularity were studied in [3]. In [10], authors investigated two numerical approaches by means of an analytical integration in the vicinity of the singular point and extraction of the singular part. A Gauss type quadrature method with a logarithmic weight function was extended to evaluate of the Cauchy type integral equations with logarithmic kernels in [30].

In this paper, we design and analyze a reliable discrete collocation technique to obtain a suitable approximate solution of (1). Our strategy is based on the following two stages. From well-known existence and uniqueness theorems [8], we can see that the first derivative of the exact solution has a singularity at the origin and behaves like as a logarithmic function. Then to establish a highly accurate approximate solution, it is necessary to represent the collocation solution as a linear combination of the suitable bases functions with logarithmic asymptotic behavior like as Müntz-logarithmic

polynomials [18]. Since in the implementation procedure, integrals with logarithmic singularities are observed to highly accurate evaluation of them, we use a generalized Gauss type quadrature with logarithmic weight function that calculates exactly integrals of polynomials [7].

The reminder of this paper is organized as follows. In the next section we present the required preliminaries for our subsequent development. Here, we introduce the Müntz-logarithmic polynomials as well as the generalized Gauss quadrature method for the integrals with logarithmic weight functions. In Section 3, we explain the application of the discrete collocation method using Müntz-logarithmic polynomials to approximate the solution of (1). In Section 4, we provide a reliable convergence analysis for the proposed algorithm that justifies the L^2-norm of the error function tends to zero as the approximation degree tends to infinity. Section 5 devotes to our numerical illustrative and Section 6 contains our conclusions.

2 Preliminaries

In this section, we give some preliminaries that are required in the sequel.

2.1 Müntz-Logarithmic polynomials

This subsection is devoted to a brief introduction on the Müntz-logarithmic polynomials. All of the details presented in this section a long with further details can be found in [18].

The Müntz-logarithmic polynomials are defined as

$$\mathcal{M}_n(x) = \mathcal{R}_n(x) + \mathcal{S}_n(x)\ln x, \quad n \geq 0,\ x \in \Omega, \tag{2}$$

where $\mathcal{R}_n(x)$ and $\mathcal{S}_n(x)$ are algebraic polynomials of degree $[\frac{n}{2}]$ and $[\frac{n-1}{2}]$, respectively; that is,

$$\mathcal{R}_n(x) = \sum_{i=0}^{[\frac{n}{2}]} r_i x^i, \qquad \mathcal{S}_n(x) = \sum_{i=0}^{[\frac{n-1}{2}]} s_i x^i.$$

It is shown that these polynomials are orthogonal with respect to the weight function $w(x) = 1$. Explicit expressions of the coefficients are obtained as follows:

Theorem 1.

- *If* $n = 2m,\ m \geq 0$, *we have*

$$\begin{cases} r_i = -\binom{m+i}{m}^2 \binom{m}{i}^2 \left[\frac{2m+1}{2i+1} + 2(m-i) \sum\limits_{j=0, j\neq i}^{m-1} \frac{2j+1}{(j-i)(j+i+1)}\right], \\ s_i = -(m-i)\binom{m+i}{m}^2 \binom{m}{i}^2, \quad 0 \leq i \leq m-1, \end{cases}$$

and

$$r_m = \binom{2m}{m}^2, \quad s_m = 0.$$

- *If* $n = 2m+1,\ m \geq 0$, *we have*

$$\begin{cases} r_i = \binom{m+i}{m}^2 \binom{m}{i}^2 \left[\frac{2m+1}{2i+1} + 2(m+i+1) \sum\limits_{j=0, j\neq i}^{m} \frac{2j+1}{(j-i)(j+i+1)}\right], \\ s_i = (m+i+1)\binom{m+i}{m}^2 \binom{m}{i}^2, \quad 0 \leq i \leq m. \end{cases}$$

Proof. See [18]. □

The first few Müntz-logarithmic polynomials are given by

$$\begin{aligned} \mathcal{M}_0(x) &= 1, \\ \mathcal{M}_1(x) &= 1 + \ln x, \\ \mathcal{M}_2(x) &= -3 + 4x - \ln x, \\ \mathcal{M}_3(x) &= 9 - 8x + 2(1+6x)\ln x, \\ \mathcal{M}_4(x) &= -11 - 24x + 36x^2 - 2(1+18x)\ln x, \\ \mathcal{M}_5(x) &= 19 + 276x - 294x^2 + 3(1+48x+60x^2)\ln x, \\ \mathcal{M}_6(x) &= -21 - 768x + 390x^2 + 400x^3 - 3(1+96x+300x^2)\ln x. \end{aligned}$$

2.2 Jacobi–Gauss quadrature

This subsection presents the mechanism of the Jacobi–Gauss quadrature [7, 9, 27], which is to seek the best numerical approximation of an integral by selecting optimal nodes at which the integrand is evaluated. The Nth-order Jacobi–Gauss quadrature formula $(\mathcal{J}G)^{\alpha,\beta}$ to approximate the integral is given by

$$\int_{-1}^{1} f(r)(1-r)^{\alpha}(1+r)^{\beta}dr \approx (\mathcal{J}G)^{\alpha,\beta}(f) := \sum_{i=1}^{N} f(x_i^{\alpha,\beta})\mathcal{W}_i^{\alpha,\beta},$$
$$\alpha,\beta > -1, \quad r \in [-1,1],$$

which the integral is evaluated exactly if $f(r)$ is a polynomial of degree $2N-1$ or less. The nodes $\{x_i^{\alpha,\beta}\}_{i=1}^{N}$ and the corresponding weights $\{\mathcal{W}_i^{\alpha,\beta}\}_{i=1}^{N}$ depend on α,β and are given by the following formulas [7, 27]:

- $\{x_i^{\alpha,\beta}\}_{i=1}^{N}$ are the zeros of the following orthonormal polynomial

$$\mathcal{P}_N^{\alpha,\beta}(r) = \sqrt{\frac{n!(2N+\alpha+\beta+1)\Gamma(N+\alpha+\beta+1)}{2^{\alpha+\beta+1}\Gamma(N+\alpha+1)\Gamma(N+\beta+1)}}J_N^{\alpha,\beta}(r),$$

  where $J_N^{\alpha,\beta}(r)$ is the classical Jacobi polynomial of degree N.

- The weights $\{\mathcal{W}_i^{\alpha,\beta}\}_{i=1}^{N}$ are given by

$$\left(\mathcal{W}_i^{\alpha,\beta}\right)^{-1} = \sum_{n=0}^{N-1}\left(\mathcal{P}_n^{\alpha,\beta}(x_i^{\alpha,\beta})\right)^2, \quad 1 \le i \le N.$$

The error term for the Nth-order Jacobi–Gauss quadrature $(\mathcal{J}G)^{\alpha,\beta}$ is given by [7]

$$\left(\int_{-1}^{1}(1-r)^{\alpha}(1+r)^{\beta}f(r)dr\right) - (\mathcal{J}G)^{\alpha,\beta}(f) = \mathcal{E}_{N,r}(\alpha,\beta,f(r)) := \delta_N^{\alpha,\beta}\frac{f^{(2N)}(\xi)}{(2N)!}, \tag{3}$$

where ξ lies somewhere on $[-1,1]$ and

$$\delta_N^{\alpha,\beta} = \frac{2^{2N+\alpha+\beta+1}N!\Gamma(N+\alpha+1)\Gamma(N+\beta+1)\Gamma(N+\alpha+\beta+1)}{(2N+\alpha+\beta+1)\Gamma(2N+\alpha+\beta+1))^2}.$$

2.3 Gauss type quadrature for logarithmic weights integrals

This subsection devotes to introduce a generalized Jacobi–Gauss quadratures [7] to approximate the following integrals

$$(a)\quad \int_{-1}^{1} f(r)\ln(1-r)(1-r)^{\alpha}dr,$$
$$(b)\quad \int_{-1}^{1} f(r)\ln(1+r)(1+r)^{\beta}dr. \tag{4}$$

Considering the first integral in (4) we have

$$\begin{aligned}\frac{\partial}{\partial\alpha}\int_{-1}^{1}(1-r)^{\alpha}f(r)dr &= \frac{\partial}{\partial\alpha}\int_{-1}^{1}e^{\alpha\ln(1-r)}f(r)dr\\ &= \int_{-1}^{1} f(r)\ln(1-r)(1-r)^{\alpha}dr,\end{aligned}$$

and equivalently

$$\begin{aligned}\int_{-1}^{1} f(r)\ln(1-r)(1-r)^{\alpha}dr &= \frac{\partial}{\partial\alpha}\int_{-1}^{1}(1-r)^{\alpha}f(r)dr\\ &\approx \frac{\partial}{\partial\alpha}\big((\mathcal{J}G)^{\alpha,0}(f)\big) = \frac{\partial}{\partial\alpha}\sum_{i=1}^{N}\left(f(x_i^{\alpha,0})\mathcal{W}_i^{\alpha,0}\right)\\ &= \sum_{i=1}^{N}\left(\frac{d\mathcal{W}_i^{\alpha,0}}{d\alpha}f(x_i^{\alpha,0}) + \mathcal{W}_i^{\alpha,0}\frac{dx_i^{\alpha,0}}{d\alpha}f'(x_i^{\alpha,0})\right)\\ &:= (G\mathcal{J}G)^{\alpha,0}(f),\end{aligned} \tag{5}$$

where "$G\mathcal{J}G$" is an abbreviation for the generalized Jacobi–Gauss. From (3), we can obtain the following formula for the error term of Nth order generalized Jacobi–Gauss quadrature $(G\mathcal{J}G)^{\alpha,0}$:

$$\begin{aligned}&\left(\int_{-1}^{1} f(r)(1-r)^{\alpha}\ln(1-r)dr\right) - (G\mathcal{J}G)^{\alpha,0}(f)\\ &\quad= \frac{\partial}{\partial\alpha}\mathcal{E}_{N,r}(\alpha,0,f(r)) := \tilde{\mathcal{E}}_{N,r}(\alpha,0,f(r))\\ &\quad:= \Bigg(\ln(2) - \frac{1}{2N+\alpha+1}\\ &\qquad + 2\Psi(N+\alpha+1) - 2\Psi(2N+\alpha+1)\Bigg)\mathcal{E}_{N,r}(\alpha,0,f(r)),\end{aligned} \tag{6}$$

where $\Psi(z) = \frac{d\ln(\Gamma(z))}{dz} = \frac{\Gamma'(z)}{\Gamma(z)}$ is the psi or digamma function.

Proceeding the same technique with (5)-(6) we can obtain the following approximation for the second integral of (4)

$$\begin{aligned}\int_{-1}^{1} f(r)\ln(1+r)(1+r)^{\beta}dr &= \frac{\partial}{\partial\beta}\int_{-1}^{1}(1+r)^{\beta}f(r)dr\\ &\approx \frac{\partial}{\partial\beta}\big((\mathcal{J}G)^{0,\beta}(f)\big) = \frac{\partial}{\partial\beta}\sum_{i=1}^{N}\left(f(x_i^{0,\beta})\mathcal{W}_i^{0,\beta}\right)\\ &= \sum_{i=1}^{N}\left(\frac{d\mathcal{W}_i^{0,\beta}}{d\beta}f(x_i^{0,\beta}) + \mathcal{W}_i^{0,\beta}\frac{dx_i^{0,\beta}}{d\beta}f'(x_i^{0,\beta})\right)\\ &:= (G\mathcal{J}G)^{0,\beta}(f),\end{aligned}$$

which has the following error function

$$\begin{aligned}&\Big(\int_{-1}^{1} f(r)(1+r)^{\beta}\ln(1+r)dr\Big) - (G\mathcal{J}G)^{0,\beta}(f)\\ &\quad= \frac{\partial}{\partial\beta}\mathcal{E}_{N,r}(0,\beta,f(r)) := \tilde{\mathcal{E}}_{N,r}(0,\beta,f(r))\\ &\quad:= \Bigg(\ln(2) - \frac{1}{2N+\beta+1}\\ &\qquad + 2\Psi(N+\beta+1) - 2\Psi(2N+\beta+1)\Bigg)\mathcal{E}_{N,r}(0,\beta,f(r)). \qquad (7)\end{aligned}$$

The relations (6) and (7) conclude that the generalized Jacobi–Gauss quadratures $(G\mathcal{J}G)^{\alpha,0}(f)$ and $(G\mathcal{J}G)^{0,\beta}(f)$ calculate the integrals of (4) exactly for polynomials of degree $2N-1$ or less same as the Jacobi–Gauss quadrature $(\mathcal{J}G)^{\alpha,\beta}(f)$.

3 Numerical approach

The main concern of this section is to obtain the discrete collocation solution of (1) when the Müntz-logarithmic polynomials are applied as the basis functions. To this end we represent the collocation solution of (1) as

$$y_N(x) = \sum_{n=0}^{N} a_n\mathcal{M}_n(x), \qquad (8)$$

such that the unknowns $\{a_n\}_{n=0}^{N}$ satisfy in the following linear algebraic system:

$$y_N(x_i) = g(x_i) + \int_0^{x_i} \ln(x_i - t) K(x_i, t) y_N(t) dt, \quad 0 \le i \le N, \tag{9}$$

where $\{x_i\}_{i=0}^{N}$ is the shifted Legendre–Gauss quadrature points [27]. Applying the variable transformation

$$t = t_i(\theta) = \frac{x_i}{2}(\theta + 1), \quad \theta \in [-1, 1],$$

we can rewrite (9) as follows:

$$\begin{aligned} y_N(x_i) &= g(x_i) + \frac{x_i}{2} \int_{-1}^{1} \ln(x_i - t_i(\theta)) K(x_i, t_i(\theta)) y_N(t_i(\theta)) d\theta \\ &= g(x_i) + \frac{x_i}{2} \Bigg\{ \ln \frac{x_i}{2} \int_{-1}^{1} \tilde{K}(x_i, \theta) y_N(t_i(\theta)) d\theta \\ &\qquad + \int_{-1}^{1} \ln(1 - \theta) \tilde{K}(x_i, \theta) y_N(t_i(\theta)) d\theta \Bigg\} \end{aligned} \tag{10}$$

for $0 \le i \le N$ and $\tilde{K}(x_i, \theta) = K(x_i, t_i(\theta))$. Substituting (8) into (10) yields

$$\sum_{n=0}^{N} a_n \left\{ \mathcal{M}_n(x_i) - \frac{x_i}{2} \left\{ \ln \frac{x_i}{2} \mathcal{A}_1^{(n,x_i)} + \mathcal{A}_2^{(n,x_i)} \right\} \right\} = g(x_i), \quad 0 \le i \le N, \tag{11}$$

where

$$\begin{aligned} \mathcal{A}_1^{(n,x_i)} &= \int_{-1}^{1} \tilde{K}(x_i, \theta) \mathcal{M}_n(t_i(\theta)) d\theta, \\ \mathcal{A}_2^{(n,x_i)} &= \int_{-1}^{1} \ln(1 - \theta) \tilde{K}(x_i, \theta) \mathcal{M}_n(t_i(\theta)) d\theta. \end{aligned}$$

Using (2), we have

$$\begin{aligned}\mathcal{A}_1^{(n,x_i)} &= \int_{-1}^{1} \tilde{K}(x_i,\theta)\Big(\mathcal{R}_n(t_i(\theta)) + \mathcal{S}_n(t_i(\theta))\ln(t_i(\theta))\Big)d\theta \\ &= \int_{-1}^{1} \tilde{K}(x_i,\theta)\Big(\mathcal{R}_n(t_i(\theta)) + \ln\frac{x_i}{2}\mathcal{S}_n(t_i(\theta))\Big)d\theta \\ &\quad + \int_{-1}^{1} \ln(1+\theta)\tilde{K}(x_i,\theta)\mathcal{S}_n(t_i(\theta))d\theta \\ &= \int_{-1}^{1} \mathcal{F}_1^{(n,x_i)}(\theta)d\theta \\ &\quad + \int_{-1}^{1} \ln(1+\theta)\mathcal{F}_2^{(n,x_i)}(\theta)d\theta,\end{aligned}$$

where

$$\begin{aligned}\mathcal{F}_1^{(n,x_i)}(\theta) &= \tilde{K}(x_i,\theta)\Big(\mathcal{R}_n(t_i(\theta)) + \ln\frac{x_i}{2}\mathcal{S}_n(t_i(\theta))\Big), \\ \mathcal{F}_2^{(n,x_i)}(\theta) &= \tilde{K}(x_i,\theta)\mathcal{S}_n(t_i(\theta)).\end{aligned} \tag{12}$$

Using quadratures $(\mathcal{J}G)^{\alpha,\beta}$ and $(G\mathcal{J}G)^{0,\beta}$, we obtain

$$\mathcal{A}_1^{(n,x_i)} \approx \mathcal{A}_{1,N}^{(n,x_i)} := (\mathcal{J}G)^{0,0}(\mathcal{F}_1^{(n,x_i)}(\theta)) + \Big[(G\mathcal{J}G)^{0,\beta}(\mathcal{F}_2^{(n,x_i)}(\theta))\Big]_{\beta=0}. \tag{13}$$

On the other hand, from (2) we can write

$$\begin{aligned}\mathcal{A}_2^{(n,x_i)} &= \int_{-1}^{1} \ln(1-\theta)\tilde{K}(x_i,\theta)\Big(\mathcal{R}_n(t_i(\theta)) + \mathcal{S}_n(t_i(\theta))\ln(t_i(\theta))\Big)d\theta \\ &= \int_{-1}^{1} \ln(1-\theta)\mathcal{F}_1^{(n,x_i)}(\theta)d\theta + \int_{-1}^{1} \ln(1-\theta)\ln(1+\theta)\mathcal{F}_2^{(n,x_i)}(\theta)d\theta \\ &= \mathcal{A}_2^{(1,n,x_i)} + \mathcal{A}_2^{(2,n,x_i)},\end{aligned} \tag{14}$$

where

$$\mathcal{A}_2^{(1,n,x_i)} = \int_{-1}^{1} \ln(1-\theta)\mathcal{F}_1^{(n,x_i)}(\theta)d\theta,$$
$$\mathcal{A}_2^{(2,n,x_i)} = \int_{-1}^{1} \ln(1-\theta)\ln(1+\theta)\mathcal{F}_2^{(n,x_i)}(\theta)d\theta.$$

Trivially we can write

$$\mathcal{A}_2^{(1,n,x_i)} \approx \mathcal{A}_{2,N}^{(1,n,x_i)} = \Big[(G\mathcal{J}G)^{\alpha,0}(\mathcal{F}_1^{(n,x_i)}(\theta))\Big]_{\alpha=0}, \tag{15}$$

and

$$\mathcal{A}_2^{(2,n,x_i)} = \int_{-1}^{0} \ln(1-\theta)\ln(1+\theta)\mathcal{F}_2^{(n,x_i)}(\theta)d\theta + \int_{0}^{1} \ln(1-\theta)\ln(1+\theta)\mathcal{F}_2^{(n,x_i)}(\theta)d\theta. \tag{16}$$

Using the variable transformation $\theta = \theta_1(s) = \frac{s+1}{2}$ in the second integral and $\theta = -\theta_1(s)$ in the first integral of (16), we obtain

$$\mathcal{A}_2^{(2,n,x_i)} = \frac{1}{2}\int_{-1}^{1} \ln\frac{1-s}{2}\ln\frac{3+s}{2}\Big(\mathcal{F}_2^{(n,x_i)}(\theta_1(s)) + \mathcal{F}_2^{(n,x_i)}(-\theta_1(s))\Big)ds$$
$$= \frac{1}{2}\left\{\int_{-1}^{1} \ln(1-s)\mathcal{F}_3^{(n,x_i)}(s)ds - \ln 2\int_{-1}^{1} \mathcal{F}_3^{(n,x_i)}(s)ds\right\},$$

where

$$\mathcal{F}_3^{(n,x_i)} = \ln\frac{3+s}{2}\Big(\mathcal{F}_2^{(n,x_i)}(\theta_1(s)) + \mathcal{F}_2^{(n,x_i)}(-\theta_1(s))\Big). \tag{17}$$

then we can write

$$\mathcal{A}_2^{(2,n,x_i)} \approx \mathcal{A}_{2,N}^{(2,n,x_i)} = \frac{1}{2}\Bigg(\Big[(G\mathcal{J}G)^{\alpha,0}(\mathcal{F}_3^{(n,x_i)}(s))\Big]_{\alpha=0} - \ln 2(\mathcal{J}G)^{0,0}(\mathcal{F}_3^{(n,x_i)}(s))\Bigg). \tag{18}$$

Substituting (18) and (15) into (14) yields

$$\mathcal{A}_2^{(n,x_i)} \approx \mathcal{A}_{2,N}^{(n,x_i)} := \mathcal{A}_{2,N}^{(1,n,x_i)} + \mathcal{A}_{2,N}^{(2,n,x_i)}. \tag{19}$$

Inserting (19) and (13) into (11), we can conclude that the discrete collocation solution of (1) is characterized by

$$\bar{y}_N(x) = \sum_{n=0}^{N} \bar{a}_n \mathcal{M}_n(x),$$

where the unknowns $\{\bar{a}_n\}_{n=0}^{N}$ satisfy in the following system of linear algebraic equations

$$\sum_{n=0}^{N} \bar{a}_n \left\{ \mathcal{M}_n(x_i) - \frac{x_i}{2} \left\{ \ln \frac{x_i}{2} \mathcal{A}_{1,N}^{(n,x_i)} + \mathcal{A}_{2,N}^{(n,x_i)} \right\} \right\} = g(x_i), \quad 0 \le i \le N, \tag{20}$$

The matrix formulation of (20) is given by

$$\underline{a}\mathcal{M} = \underline{g}, \tag{21}$$

where $\underline{a} = [\bar{a}_0, \bar{a}_1, \ldots, \bar{a}_N]$, $\underline{g} = [g(x_0), g(x_1), \ldots, g(x_N)]^T$, and

$$\mathcal{M} = \mathcal{M}_1 - (\mathcal{A}_1\mathcal{D}_1 + \mathcal{A}_2\mathcal{D}_2),$$

such that

$$\mathcal{M}_1 = \begin{pmatrix} \mathcal{M}_0(x_0) & \mathcal{M}_0(x_1) & \cdots & \mathcal{M}_0(x_N) \\ \mathcal{M}_1(x_0) & \mathcal{M}_1(x_1) & \cdots & \mathcal{M}_1(x_N) \\ \vdots & \vdots & \vdots & \vdots \\ \mathcal{M}_N(x_0) & \mathcal{M}_N(x_1) & \cdots & \mathcal{M}_N(x_N) \end{pmatrix},$$

$$\mathcal{A}_1 = \begin{pmatrix} \mathcal{A}_{1,N}^{(0,x_0)} & \mathcal{A}_{1,N}^{(0,x_1)} & \cdots & \mathcal{A}_{1,N}^{(0,x_N)} \\ \mathcal{A}_{1,N}^{(1,x_0)} & \mathcal{A}_{1,N}^{(1,x_1)} & \cdots & \mathcal{A}_{1,N}^{(1,x_N)} \\ \vdots & \vdots & \vdots & \vdots \\ \mathcal{A}_{1,N}^{(N,x_0)} & \mathcal{A}_{1,N}^{(N,x_1)} & \cdots & \mathcal{A}_{1,N}^{(N,x_N)} \end{pmatrix},$$

$$\mathcal{A}_2 = \begin{pmatrix} \mathcal{A}_{2,N}^{(0,x_0)} & \mathcal{A}_{2,N}^{(0,x_1)} & \cdots & \mathcal{A}_{2,N}^{(0,x_N)} \\ \mathcal{A}_{2,N}^{(1,x_0)} & \mathcal{A}_{2,N}^{(1,x_1)} & \cdots & \mathcal{A}_{2,N}^{(1,x_N)} \\ \vdots & \vdots & \vdots & \vdots \\ \mathcal{A}_{2,N}^{(N,x_0)} & \mathcal{A}_{2,N}^{(N,x_1)} & \cdots & \mathcal{A}_{2,N}^{(N,x_N)} \end{pmatrix},$$

and $\mathcal{D}_1$ and $\mathcal{D}_2$ are the diagonal matrices with diagonal entries

$$(\mathcal{D}_1)_{i,i} = \frac{x_i}{2}\ln\frac{x_i}{2}, \quad (\mathcal{D}_2)_{i,i} = \frac{x_i}{2}, \quad 0 \le i \le N.$$

4 Convergence analysis

In this section, we provide a reliable error analysis for the proposed technique to justify convergence of the proposed approach. In our analysis we shall apply the following definitions and lemmas.

Definition 1.[[1], [19–23], [27]]

1. $L^2(\Omega) = \{u| \ \ \|u\|_2^2 := \int\limits_{\Omega} |u(x)|^2 dx < \infty\}$.
2. $C(\Omega)$ is the space of all continuous functions on Ω.
3. $I_N u$ is the Legendre–Gauss interpolation polynomial and defines by

$$I_N u(x) = \sum_{i=0}^{N} u(x_i) L_i(x),$$

   where $L_i(x)$, $0 \le i \le N$, are the Lagrange interpolation basis functions associated with the Legendre–Gauss points $\{x_i\}_{i=0}^N$.
4. Let $(\mathcal{X}, \|.\|_{\mathcal{X}})$ and $(\mathcal{Y}, \|.\|_{\mathcal{Y}})$ be normed vector spaces, and let $\mathcal{K} : \mathcal{X} \to \mathcal{Y}$ be a linear operator. Then $\mathcal{K}$ is compact, if the set

$$\{\mathcal{K}u| \ \|u\|_{\mathcal{X}} \le 1\}$$

   has compact closure in $\mathcal{Y}$. For example, the integral operators with continuous and weakly singular kernels are compact operators on $C(\Omega)$ and $L^2(\Omega)$.

Lemma 1.[see [8]] *Let $K(x,t) \in C(\Omega \times \Omega)$; then for any $g(x) \in C(\Omega)$, the Volterra type weakly singular integral equation with logarithmic kernel (1) possesses a unique solution $y(x) \in C(\Omega)$.*

Lemma 2.[see [1]] *Let $\mathcal{X}$ be a Banach space, and let $\mathcal{K} : \mathcal{X} \to \mathcal{X}$ be compact. Then the equation $(\lambda - \mathcal{K})u = f, \ \ \lambda \neq 0$ has a unique solution $u \in \mathcal{X}$ if and only if the homogeneous equations $(\lambda - \mathcal{K})z = 0$ have only the trivial solution $z = 0$. In this case the operator $\lambda - \mathcal{K} : \mathcal{X} \to \mathcal{X}$ has a bounded inverse $(\lambda - \mathcal{K})^{-1}$.*

Lemma 3.[see [26]] *If $u(x) \in C(\Omega)$, then we have*

$$\|u - I_N u\|_2 \to 0, \quad as \quad N \to \infty.$$

Lemma 4.[see [19]] *For every bounded function $u(x)$, there exists a constant C independent of $u(x)$ such that*

$$\sup_N \|I_N u\|_2 \leq C\|u\|_\infty.$$

Theorem 2. *Under assumptions of Lemma 1, assume that the following conditions are satisfied:*

1. *The given functions $g(x), K(x,t)$, and the approximate solution $\bar{y}_N(x)$ are continuous on their domains.*
2. *The homogeneous equation (1) with $g(x) = 0$ has the only trivial solution.*
3. *$\bar{y}_N(x) \in C(\Omega)$ and the numerical integral operator*

$$\mathcal{K}_N \bar{y}_N(x) := \sum_{n=0}^{N} \bar{a}_n \ \frac{x}{2}\Big\{ \ln \frac{x}{2} \mathcal{A}_{1,N}^{n,x} + \mathcal{A}_{2,N}^{n,x} \Big\}$$

is a bounded operator on $C(\Omega)$ to $C(\Omega)$ with $\mathcal{A}_{1,N}^{n,x}$ and $\mathcal{A}_{2,N}^{n,x}$ defined in (13) and (19), respectively.

4. *The quadrature errors*

$$\|\mathcal{E}_{N,\theta}(0,0,\mathcal{F}_1^{n,x}(\theta))\|_\infty, \|\mathcal{E}_{N,s}(0,0,\mathcal{F}_3^{n,x}(s))\|_\infty,$$

$$\Big\| \Big[\tilde{\mathcal{E}}_{N,\theta}(\alpha,0,\mathcal{F}_1^{n,x}(\theta))\Big]_{\alpha=0} \Big\|_\infty, \Big\| \Big[\tilde{\mathcal{E}}_{N,\theta}(0,\beta,\mathcal{F}_2^{n,x}(\theta))\Big]_{\beta=0} \Big\|_\infty,$$

$$\Big\| \Big[\tilde{\mathcal{E}}_{N,s}(\alpha,0,\mathcal{F}_3^{n,x}(s))\Big]_{\alpha=0} \Big\|_\infty, \qquad 0 \leq n \leq N,$$

converge to zero as $N \to \infty$. Here the functions $\mathcal{F}_1^{n,x}(\theta), \mathcal{F}_2^{n,x}(\theta)$ and $\mathcal{F}_3^{n,x}(s)$ are given in (12) and (17), respectively, and the error terms $\mathcal{E}_{N,r}(0,0,f(r))$,
$\tilde{\mathcal{E}}_{N,r}(\alpha,0,f(r)), \tilde{\mathcal{E}}_{N,r}(0,\beta,f(r))$ are defined in (3), (6), and (7), respectively.

Then we have

$$\lim_{N\to\infty} \|y(x) - \bar{y}_N(x)\|_2 = 0.$$

Proof. Using (20), we can conclude that the discrete collocation solution $\bar{y}_N(x)$ for the equation (1) satisfies in the following operator system:

$$\bar{y}_N(x_i) = g(x_i) + \mathcal{K}_N \bar{y}_N(x_i), \qquad 0 \leq i \leq N, \tag{22}$$

where the operator $\mathcal{K}_N$ is the numerical integral operator defined in the assumption 3. Multiplying $L_i(x)$ on both sides of (22) and summing up from $i = 0$ to $i = N$ yield

$$I_N(\bar{y}_N) = I_N g + I_N \mathcal{K}_N \bar{y}_N. \tag{23}$$

Subtracting (1) from (23) gives

$$y - I_N(\bar{y}_N) = (g - I_N g) + (\mathcal{K}y - I_N \mathcal{K}_N \bar{y}_N),$$

or equivalently

$$(\mathrm{id} - \mathcal{K})\bar{e}_N = e_{I_N}(g) - e_{I_N}(\bar{y}_N) + e_{I_N}(\mathcal{K}\bar{y}_N) + I_N\big(\mathcal{K}\bar{y}_N - \mathcal{K}_N\bar{y}_N\big), \tag{24}$$

where $\bar{e}_N = y(x) - \bar{y}_N(x)$ is the discrete collocation error function, $e_{I_N}u = u - I_N u$ is the interpolation error function, id is the identity operator, and $\mathcal{K}$ is the following continuous and compact integral operator with weakly singular logarithmic kernel

$$\mathcal{K}y(x) = \int_0^x \ln(x-t)K(x,t)y(t)dt.$$

Applying Lemma 2 with $\mathcal{X} = C(\Omega)$ and using the assumption 2, the relation (24) can be rewritten as

$$\|\bar{e}_N\|_2 \le \|(\mathrm{id} - \mathcal{K})^{-1}\|_\infty \Big(\|e_{I_N}(g)\|_2 + \|e_{I_N}(\bar{y}_N)\|_2 + \|e_{I_N}(\mathcal{K}\bar{y}_N)|_2 + \|I_N\big(\mathcal{K}\bar{y}_N - \mathcal{K}_N\bar{y}_N\big)\|_2 \Big), \tag{25}$$

with $\|(\mathrm{id} - \mathcal{K})^{-1}\|_\infty < \infty$. Due to Lemma 3 and assumption 1 we have

$$\|e_{I_N}(g)\|_2, \quad \|e_{I_N}(\bar{y}_N)\|_2, \quad \|e_{I_N}(\mathcal{K}\bar{y}_N)|_2 \to 0, \quad \text{as} \quad N \to \infty. \tag{26}$$

Now, it is sufficient that we show $\|I_N\big(\mathcal{K}\bar{y}_N - \mathcal{K}_N\bar{y}_N\big)\|_2 \to 0$ as $N \to \infty$. To this end, using Lemma 4 and assumption 3, we can write

$$\|I_N\big(\mathcal{K}\bar{y}_N - \mathcal{K}_N\bar{y}_N\big)\|_2 \le C_1 \|\mathcal{K}\bar{y}_N - \mathcal{K}_N\bar{y}_N\|_\infty, \tag{27}$$

which $\mathcal{K}\bar{y}_N - \mathcal{K}_N\bar{y}_N$ is the numerical integration error function. According to the definition of $\mathcal{K}_N$ and numerical approach proposed in the previous section, we can deduce that the numerical integration error is established by calculating the errors obtained from the applying Jacobi–Gauss and generalized Jacobi–Gauss formulas in (13), (15), (18), and (19). Consequently, we can write (27) as follows:

$$\|I_N\big(\mathcal{K}\bar{y}_N - \mathcal{K}_N\bar{y}_N\big)\|_2 \le C_2 \sum_{n=0}^{N}\Bigg(\|\mathcal{E}_{N,\theta}(0,0,\mathcal{F}_1^{n,x}(\theta))\|_\infty + \Big\|\Big[\tilde{\mathcal{E}}_{N,\theta}(0,\beta,\mathcal{F}_2^{n,x}(\theta))\Big]_{\beta=0}\Big\|_\infty + \Big\|\Big[\tilde{\mathcal{E}}_{N,\theta}(\alpha,0,\mathcal{F}_1^{n,x}(\theta))\Big]_{\alpha=0}\Big\|_\infty + \Big\|\Big[\tilde{\mathcal{E}}_{N,s}(\alpha,0,\mathcal{F}_3^{n,x}(s))\Big]_{\alpha=0}\Big\|_\infty + \|\mathcal{E}_{N,s}(0,0,\mathcal{F}_3^{n,x}(s))\|_\infty\Bigg),$$

and thereby using assumption 4 in the inequality above, we deduce

$$\|I_N\big(\mathcal{K}\bar{y}_N - \mathcal{K}_N\bar{y}_N\big)\|_2 \to 0, \quad \text{as} \quad N \to 0. \tag{28}$$

Finally, the required result can be obtained by applying (26) and (28) in (25). □

5 Numerical Results

In this section, we illustrate some examples using the method proposed in the previous section and confirm its validity. All of calculations performed on a PC running *Mathematica* software. In the obtained results we presented some essential items regarding the L^2-norms of the error functions and comparison results between our scheme and the well known Chebyshev collocation method [9, 27].

Example 1. Consider the following problem

$$y(x) = g(x) + \int_0^x \ln(x-t)y(t)dt$$

with

$$g(x) = x(\ln x - 1) + \frac{x^2}{12}\Big(\pi^2 - 21 + 18\ln x - 6\ln^2 x\Big)$$

and $y(x) = x(\ln x - 1)$ as the exact solution.

We solve this problem by the proposed method with $N = 3$. Indeed, we seek a discrete collocation solution in the form

$$\bar{y}_3(x) = \sum_{n=0}^{3} \bar{a}_n \mathcal{M}_n(x),$$

to approximate this problem and find its unknown coefficients such that they satisfy in the linear system (21) with $N = 3$. Considering the Gauss–Legendre collocation points

$$x_0 = 0.330009, \quad x_1 = 0.669991, \quad x_2 = 0.0694318, \quad x_3 = 0.930568,$$

and the operational matrices

$$\mathcal{M}_1 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ -0.1086 & 0.5995 & -1.6674 & 0.9280 \\ -0.5713 & 0.0805 & -0.0549 & 0.7942 \\ -0.2477 & -0.3808 & 0.8873 & 0.6080 \end{pmatrix},$$

$$\mathcal{A}_1 = \begin{pmatrix} 2 & 2 & 2 & 2 \\ -2.2173 & -0.8009 & -5.3348 & -0.1439 \\ -0.4627 & -0.5191 & 1.6126 & -0.1338 \\ 0.5550 & -0.2017 & 0.1359 & -0.1194 \end{pmatrix},$$

$$\mathcal{A}_2 = \begin{pmatrix} -0.6137 & -0.6137 & -0.6137 & -0.6137 \\ -0.6095 & -1.0441 & 0.3471 & -1.2457 \\ 0.7718 & 0.1092 & 0.6562 & -0.5302 \\ 0.1912 & 0.6068 & -1.3532 & 0.0376 \end{pmatrix},$$

$$\mathcal{D}_1 = \begin{pmatrix} -0.2973 & 0 & 0 & 0 \\ 0 & -0.3664 & 0 & 0 \\ 0 & 0 & -0.1167 & 0 \\ 0 & 0 & 0 & -0.3560 \end{pmatrix},$$

$$\mathcal{D}_2 = \begin{pmatrix} 0.1650 & 0 & 0 & 0 \\ 0 & 0.3350 & 0 & 0 \\ 0 & 0 & 0.0247 & 0 \\ 0 & 0 & 0 & 0.4653 \end{pmatrix},$$

the system of linear algebraic equations (21) is presented in the following form

$$\begin{cases} 1.0449 + 1.6959\bar{a}_0 - 0.6673\bar{a}_1 - 0.8362\bar{a}_2 - 0.1142\bar{a}_3 = 0, \\ 1.6603 + 1.9383\bar{a}_9 + 0.6558\bar{a}_1 - 0.1463\bar{a}_2 - 0.6580\bar{a}_3 = 0, \\ 0.2955 + 1.2546\bar{a}_0 - 2.3019\bar{a}_1 + 0.1105\bar{a}_2 + 0.9501\bar{a}_3 = 0, \\ 1.8965 + 1.9975\bar{a}_0 + 1.4564\bar{a}_1 + 0.9933\bar{a}_2 + 0.5830\bar{a}_3 = 0, \end{cases}$$

which its solution gives by

$$\bar{a}_0 = -0.75, \quad \bar{a}_1 = -0.25, \quad \bar{a}_2 = -0.0833, \quad \bar{a}_3 = 0.0833.$$

Consequently the discrete collocation solution $\bar{y}_3(x)$ represents by

$$\bar{y}_3(x) = -5.2953 \times 10^{-7} - x - 1.3027 \times 10^{-7} \ln x + 0.9999 x \ln x.$$

The L^2-norm of the error function is 1.3445×10^{-13} that is in a very good agreement with the exact ones whereas the approximation degree is very small($N = 3$).

Example 2. Consider the following problem;

$$y(x) = g(x) + \int_0^x \ln(x-t) e^{x+t} y(t) dt,$$

with

$$g(x) = e^{-x} \ln x + \frac{x e^x}{6} \Big(-12 + \pi^2 - 6 \ln x (-2 + \ln x) \Big),$$

and $y(x) = e^{-x} \ln x$, as the exact solution.

We solve the problem and report the obtained results in Table 1 and Figure 1. In Figure 1, we plot the L^2-norm of the error function in terms of the various values of the degree of approximation N. Figure 1 shows that the proposed algorithm obtains a good accuracy with suitable values of N. Moreover to make a comparison, we also solve this problem by implementing the well-known Chebyshev collocation method [9, 27] and give the obtained results in Table 1. Based on Table 1, we confirm that the Müntz-logarithmic polynomials makes faster rate of convergence for the discrete collocation solution of this problem compared with the classical Chebyshev polynomials.

Example 3. Consider the following problem:

$$y(x) = g(x) + x \int_0^x \ln(x-t) t^2 y(t) dt,$$

with

$$g(x) = x^{\frac{5}{2}} - \frac{2x^{\frac{13}{2}}(-13016 + 6930 \ln 2 + 3465 \ln x)}{38115},$$

Table 1: The numerical errors of Example 2

N	Numerical Errors	
	Our Method	Chebyshev collocation Method [9, 27]
2	3.61×10^{-3}	1.3×10^{-1}
4	8.13×10^{-6}	4.22×10^{-2}
6	9.38×10^{-8}	2.14×10^{-2}
8	9.84×10^{-10}	1.3×10^{-2}
10	3.44×10^{-12}	8.73×10^{-3}
12	6.37×10^{-15}	6.28×10^{-3}

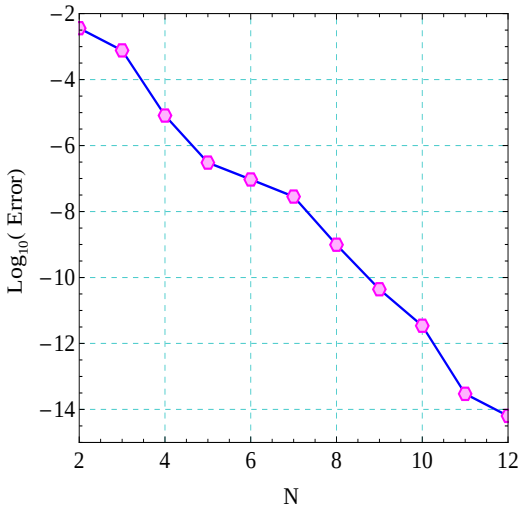


Figure 1: Obtained L^2- norm of the error function versus N for Example 2

and $y(x) = x^2\sqrt{x}$ as the exact solution.

We have calculated the approximate solution with different values of N and displayed the obtained results in Table 2 and Figure 2. Table 2 and Figure 2, present the L^2-norm of the error functions versus N. As it can be observed although the exact solution has singularity at zero, the numerical errors are decreased with an appropriate rate as the approximation degree N is increased.

Example 4. Consider the following problem:

$$y(x) = g(x) + \int_0^x \ln(x-t)y(t)dt$$

Table 2: The numerical errors of Example 3

N	Numerical Errors
2	2.69×10^{-2}
4	3.32×10^{-4}
6	1.18×10^{-7}
8	1.4×10^{-10}
10	1.84×10^{-12}
12	5.43×10^{-15}

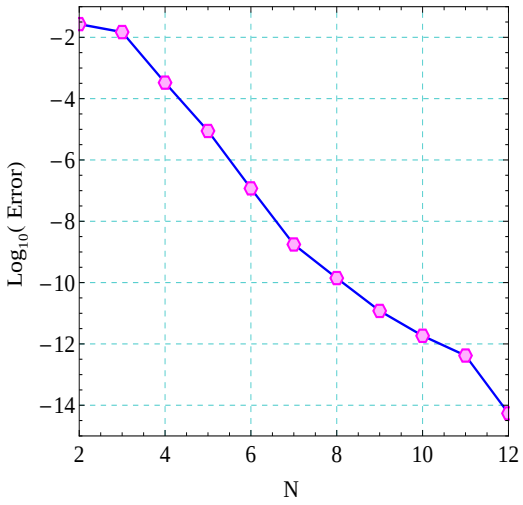


Figure 2: Obtained L^2- norm of the error function versus N for Example 3

with

$$g(x) = e^x\big(1 + \gamma + \Gamma(0, x)\big) + \ln x,$$

where γ is Euler's constant with the numerical value$\simeq 0.577216$ and $\Gamma(0, x)$ is incomplete gamma function satisfies

$$\Gamma(a, z) = \int_z^{\infty} t^{a-1} e^{-t} dt.$$

Here the exact solution is given by $y(x) = e^x$.

The obtained numerical results from implementation of the proposed discrete collocation scheme are reported in Table 3 and Figure 3. The presented

Table 3: The numerical errors of Example 4

N	Numerical Errors
2	6.08×10^{-3}
4	2.01×10^{-4}
6	2.68×10^{-6}
8	1.78×10^{-8}
10	6.99×10^{-11}
12	1.85×10^{-13}

results confirm that our scheme provides reliable results for smooth solutions of (1).

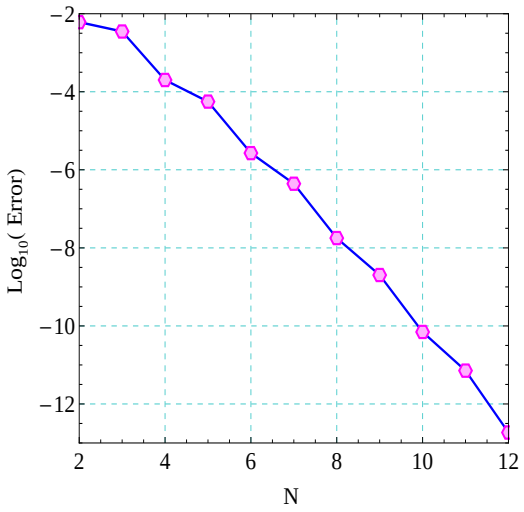


Figure 3: Obtained L^2- norm of the error function versus N for Example 4

6 Conclusion

In this article, we developed a new discrete collocation method based on the Müntz-logarithmic polynomials as basis functions. Moreover, we used highly accurate Jacobi–Gauss and generalized Jacobi–Gauss quadratures to approximate the integrals with Jacobi and logarithmic weights, respectively. Convergence analysis of the proposed method were presented and some numerical examples were illustrated to confirm the applicability of the presented discrete collocation scheme.

References

1. Atkinson, K. E. *The Numerical Solution of Integral Equations of the Second Kind,* Cambridge, 1997.

2. Andreasen, M. G. and Mei, K. K. Comments on "Scattering by conducting rectangular cylinders", *IEEE Trans, Antennas and Propagation,* 11 (1963), 52–56.

3. Andronov, I. V. Integro-differential equations of the convolution on a finite interval with kernel having a logarithmic singularity, *J. Math. Sci.* 79 (1996), no. 4, 1161–1165.

4. Assari, P., Adibi, H., and Dehghan, M. A meshless discrete Galerkin(MDG) method for the numerical solution of integral equations with logarithmic kernels, *J. Comput. Appl. Math,* 267 (2014), 160–181.

5. Banaugh, P. P. and Goldsmith, W. Diffraction of steady acoustic waves by surfaces of arbitrary shape, *J. Acoust. Soc. Am.,* 35 (1963), 1590–1601.

6. Banaugh, P. P. and Goldsmith, W. Diffraction of steady elastic waves by surfaces of arbitrary shape, *J. Appl. Mech,* 30 (1963), 589–597.

7. Ball, J. S. and Beebe, N. H. F. Efficient Gauss-related quadrature for two classes of logarithmic weight functions, *ACM Transactions on Mathematical Software,* 33 (2007), no. 3, Article No. 19.

8. Brunner, H. *Collocation Methods for Volterra and Related Functional Equations,* Cambridge University Press: Cambridge, 2004.

9. Canuto, C., Hussaini, M. Y.,Quarteroni, A., and Zang, T. A. *Spectral Methods, Fundamentals in Single Domains,* Berlin: Springer-Verlag, 2006.

10. Christiansen S. Numerical solution of an integral equation with a logarithmic kernel, *BIT,* 11 (1971), 276–287.

11. Domínquez, V. High-order collocation and quadrature methods for some logarithmic kernel integral equations on open arcs, *J. Comput. Appl. Math.,* 161 (2003), 145–159.

12. Guseinov, E. A. and Il′inskii, A. S.Integral equations of the first kind with a logarithmic singularity in the kernel and their application in problems of diffraction by thin screens, *U. S. S. R. Comput. Maths. Math. Phys.,* 27 (1987), no. 4, 58–63.

13. Gusenkova,A. A. and Pleshchinskii, N. B. Integral equations with logarithmic singularities in the kernels of boundary-value problems of plane elasticity theory for regions with a defect, *J. Appl. Maths. Mechs.,* 64 (2000), no. 3, 433–441.

14. Khader, A. H., Shamardan, A. B., Callebaut, D. K. and Sakran, M. R. A. Solving integral equations with logarithmic kernels by Chebyshev polynomials, *Numer. Algorithms,* 47 (2008), 81–93.

15. Mei, K. K. and Van Bladel, J. G. Low-frequency scattering by rectangular cylinders, *IEEE Trans, Antennas and Propagation,* 11 (1963), 52–56.

16. Mei, K. K. and Van Bladel, J. G. Scattering by perfectly-conducting rectangular cylinders , *IEEE Trans, Antennas and Propagation,* 11 (1963), 185–192.

17. Mennouni, A.Airfoil polynomials for solving integro-diferential equations with logarithmic kernel, *Appl. Math. Comput.,* 218 (2012), 11947–11951.

18. Milovanović, G. V. Müntz orthogonal polynomials and their numerical evaluation, *Internat. Ser. Numer. Math.,* 131 (1999), Birkhäuser Verlag 13asel/ Switzerland.

19. Mokhtary, P.Reconstruction of exponentially rate of convergence to Legendre collocation solution of a class of fractional integro-differential equations, *J. Comput. Appl. Math.*, 279 (2015), 145–158.

20. Mokhtary, P.Numerical treatment of a well-posed Chebyshev Tau method for Bagley-Torvik equation with high-order of accuracy, *Numer. Algorithms*, 72 (2016), 875–891.

21. Mokhtary, P.Discrete Galerkin method for fractional integro-differential equations, *Acta. Math. Sci.*, 36 (2016), no. B(2), 560–578.

22. Mokhtary, P. Numerical analysis of an operational Jacobi Tau method forfractional weakly singular integro-differential equations, *Appl. Numer. Math.*, 121 (2017), 52–67.

23. Mokhtary, P. and Ghoreishi, F. Convergence analysis of spectral Tau method for fractional Riccati differential equations, *Bull. Iranian Math. Soc.*, 40 (2014), no. 5, 1275–1296.

24. Muschelischwili, N. I. *Singuläre Integralgleichungen,* Akademie-Verlag, Berlin, 1965.

25. Noble, B. Integral equation perturbation methods in low frequency diffraction in R. E. Langer electromagnetic waves, *The University of Wisconsin Press, Madison* (1963), 323–360.

26. Rivlin, T. J. *An Introduction to the Approximation of Functions*, United States, 1969.

27. Shen, J., Tang, T., and Wang ,L.-L., *Spectral Methods, Algorithms, Analysis and Applications,* Springer, 2011.

28. Symm, G. T. An integral equation method in conformed mapping, *Numer. Math.,* 9 (1966), 250–258.

29. Tang, T., McKee, S., and Diogo, T. Product integration methods for an integral equation with logarithmic singular kernel, *Appl. Numer. Math.,* 9 (1992), 259–266.

30. Theocaris, P. S., Chrysakis, A. C., and Ioakimidis,N. I. Cauchy-type integrals and integral equations with logarithmic singularities, *J. Eng. Math.,* 13 (1979), no. 1, 63–74.

Iranian Journal of Numerical Analysis and Optimization
Vol. 8, No. 2, (2018), pp 119–140
DOI:10.22067/ijnao.v8i2.64202

An efficient hybrid algorithm based on genetic algorithm (GA) and Nelder–Mead (NM) for solving nonlinear inverse parabolic problems

H. Dana Mazraeh* and R. Pourgholi

Abstract

In this paper a hybrid algorithm based on genetic algorithm (GA) and Nelder–Mead (NM) simplex search method is combined with least squares method for the determination of temperature in some nonlinear inverse parabolic problems (NIPP). The performance of hybrid algorithm is established with some examples of NIPP. Results show that hybrid algorithm is better than GA and NM separately. Numerical results are obtained by implementation expressed algorithms on 2.20GHz clock speed CPU.

Keywords: Hybrid; NIPP; Nonlinear inverse parabolic problem; Genetic algorithm; Nelder–Mead; The least squares method.

1 Introduction

Most phenomena in real world are described through nonlinear equations, and these type of equations have attracted lots of attention among scientists. For example, consider that the sun shines in a room, and it begins to warm up. Apparently, the physical property here is the temperature, which is a function of the time and space, and the governing equations that represent the temperature variations, are a nonlinear parabolic problem (NPP) [1].

*Corresponding author
Received 5 May 2017; revised 25 November 2017; accepted 13 June 2018
H. Dana Mazraeh
School of Mathematics and Computer Science, Damghan University, P.O.Box 36715-364, Damghan, Iran. e-mail:dana@du.ac.ir

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

In linear and nonlinear parabolic problems, we are usually facing a problem, where problem conditions, initial conditions, and boundary conditions are identified and in the main equation only the equation main function is unknown. In fact, there is just one unknown factor at the problem. These problems are called direct problems.

On the contrary there is another category of problems. In this category, in addition to the main equation of the problem, there are another unknown parts such as boundary conditions. This type of problems are called inverse problems [12].

In the present paper, we consider nonlinear inverse parabolic problems as the following form:

$$U_t = \phi(x,t,U,U_x,U_{xx}), \quad (x,t) \in (0,1) \times (0,t_m), \tag{1a}$$

with the initial condition

$$U(x,0) = f(x), \quad x \in [0,1], \tag{1b}$$

and the boundary conditions

$$U(0,t) = p(t), \quad t \in [0,t_m], \tag{1c}$$
$$U(1,t) = q(t), \quad t \in [0,t_m], \tag{1d}$$

and the overspecified condition

$$U(\alpha,t) = s(t), \quad t \in [0,t_m], \quad 0 < \alpha < 1, \tag{1e}$$

where ϕ is some nonlinear expression in terms of U, U_x, U_{xx}, and $f(x)$ is a continuous known function. $p(t)$ and $q(t)$ are infinitely differentiable known functions, and t_M represents the final existence time for the time evolution of the problem, while function $q(t)$ is unknown, which remains to be determined from some interior temperature measurements (overspecified condition).

In this paper we present a hybrid algorithm based on Nelder–Mead simplex search method and genetic algorithm for solving NIPPs. Genetic algorithm is one of the basic heuristics to solve various types of problems. In recent years, this algorithm has been applied on some linear and nonlinear partial differential equation to find unknown term in these equations. On the other hand, there are some hybrid algorithms, which use local search method as an inner subroutine to reduce the likelihood of the premature convergence and to expedite convergence rate of genetic algorithms. These algorithms are called "*Memetic Algorithms*" or "*Hybrid Algorithms*". In this study, to expiate convergence rate of genetic algorithm and increase accuracy of solutions, we add Nelder–Mead method to the algorithm as a local search method. The Nelder–Mead method is a commonly applied numerical method used to find the minimum or maximum of an objective function in a multidimensional

space. It is successfully applied to nonlinear optimization problems for which derivatives may not be known. However, the Nelder–Mead technique is a heuristic search method that can converge to nonstationary points [16] on problems that can be solved by alternative methods.

Our main purpose is to find an unknown boundary condition in NIPPs using our proposed hybrid method. Because by having this unknown boundary condition, problem becomes a direct NPP and fortunately, many methods have been reported to solve direct NPPs. In this paper we use finite difference method for solving direct NPPs.

2 Some nonlinear inverse parabolic problems

In this section, we consider three NIPPs. The first equation is formulated as follows [1]:

$$U_t(x,t) + U(x,t)U_x(x,t) = U_{xx}(x,t), \quad 0 < x < 1, \quad 0 < t < t_M, \tag{2a}$$
$$U(x,0) = f(x), \quad 0 \le x \le 1, \tag{2b}$$
$$U(0,t) = p(t), \quad 0 \le t \le t_M, \tag{2c}$$
$$U(1,t) = q(t), \quad 0 \le t \le t_M, \tag{2d}$$

and the overspecified condition

$$U(\alpha,t) = s(t), \quad 0 \le t \le t_M, \quad 0 < \alpha < 1, \tag{2e}$$

The second equation is formulated as follows:

$$U_t(x,t) - U(x,t)(1 - U(x,t)) = U_{xx}(x,t), \quad 0 < x < 1, \quad 0 < t < t_M, \tag{3a}$$
$$U(x,0) = f(x), \quad 0 \le x \le 1, \tag{3b}$$
$$U(0,t) = p(t), \quad 0 \le t \le t_M, \tag{3c}$$
$$U(1,t) = q(t), \quad 0 \le t \le t_M, \tag{3d}$$

and the overspecified condition

$$U(\alpha,t) = s(t), \quad 0 \le t \le t_M, \quad 0 < \alpha < 1, \tag{3e}$$

and the third equation is formulated as follows:

$$U_t(x,t) = a(t)U_{xx}(x,t), \quad 0 < x < 1, \quad 0 < t < t_M, \tag{4a}$$
$$U(x,0) = f(x), \quad 0 \le x \le 1, \tag{4b}$$
$$U(0,t) = p(t), \quad 0 \le t \le t_M, \tag{4c}$$
$$U(1,t) = q(t), \quad 0 \le t \le t_M, \tag{4d}$$

and the overspecified condition

$$U(\alpha,t)=s(t), \qquad 0\le t\le t_M, \quad 0<\alpha<1, \tag{4e}$$

where $f(x)$ is a continuous known function, $p(t)$, $a(t)$, and $s(t)$ are infinitely differentiable known functions and t_M represents the final existence time for the time evolution of the problem. In these equations, if $q(t)$ is unknown, then we are facing nonlinear inverse parabolic problems. In this case, unknown $q(t)$ must be determined from some interior temperature measurements(Overspecified condition).

Remark 1. In this study we use implicit finite difference approximation (Crank–Nicolson method) for discretizing above equations.

Therefore we have the following discretization for the first equation:

$$\begin{aligned}-r_1U_{i-1,j+1}+(2+2r_1)U_{i,j+1}-r_1U_{i+1,j+1}&\\ =r_1U_{i-1,j}+(2-2r_1)U_{i,j}+r_1U_{i+1,j}&\\ +2r_2({U_{i,j}}^2-U_{i,j}U_{i+1,j}),\ i=1,\dots,N-1,\ j=0,\dots,N-1,&\end{aligned} \tag{5}$$

$$\begin{aligned}U_{i,0}&=f(ih), \quad j=0,\ i=1,\dots,N-1,\\ U_{0,j}&=p(jk), \quad i=0,\ j=0,1,\dots,N-1,\\ U_{N,j}&=q(jk), \quad i=N,\ j=0,1,\dots,N-1,\end{aligned}$$

where $x_i=ih$, $t_j=jk$, $r_1=k/h^2$, and $r_2=k/h$.

Using equation (5), we obtain the following linear algebraic system of equations:

$$\begin{pmatrix} 2+2r_1 & -r_1 & 0 & 0 & 0 & 0 & 0\\ -r_1 & 2+2r_1 & -r_1 & 0 & 0 & 0 & 0\\ . & . & . & . & . & . & .\\ . & . & . & . & . & . & .\\ . & . & . & . & . & . & .\\ 0 & 0 & 0 & 0 & -r_1 & 2+2r_1 & -r_1\\ 0 & 0 & 0 & 0 & 0 & -r_1 & 2+2r_1 \end{pmatrix} \begin{pmatrix} U_{1,j+1}\\ U_{2,j+1}\\ .\\ .\\ .\\ U_{N-2,j+1}\\ U_{N-1,j+1} \end{pmatrix}$$

$$= \begin{pmatrix} 2-2r_1 & r_1 & 0 & 0 & 0 & 0 & 0\\ r_1 & 2-2r_1 & r_1 & 0 & 0 & 0 & 0\\ . & . & . & . & . & . & .\\ . & . & . & . & . & . & .\\ . & . & . & . & . & . & .\\ 0 & 0 & 0 & 0 & r_1 & 2-2r_1 & r_1\\ 0 & 0 & 0 & 0 & 0 & r_1 & 2-2r_1 \end{pmatrix} \begin{pmatrix} U_{1,j}\\ U_{2,j}\\ .\\ .\\ .\\ U_{N-2,j}\\ U_{N-1,j} \end{pmatrix}$$

$$+r_1\begin{pmatrix} U_{0,j}+U_{0,j+1} \\ 0 \\ . \\ . \\ . \\ 0 \\ U_{N,j}+U_{N,j+1} \end{pmatrix} + 2r_2\begin{pmatrix} {U_{1,j}}^2 - U_{1,j}U_{2,i} \\ {U_{2,j}}^2 - U_{2,j}U_{3,i} \\ . \\ . \\ . \\ {U_{N-2,j}}^2 - U_{N-2,j}U_{N-1,i} \\ {U_{N-1,j}}^2 - U_{N-1,j}U_{N,i} \end{pmatrix} \tag{6}$$

Linear system (6) gives $(N-1)$ unknown pivotal values along the boundary $x = 1$.
We have the following discretization for the second equation:

$$\begin{aligned} -rU_{i-1,j+1} + (2+2r)U_{i,j+1} - rU_{i+1,j+1} \\ = rU_{i-1,j} + (2-2r)U_{i,j} \\ + rU_{i+1,j} + 2k(U_{i,j} - {U_{i,j}}^2),\ i = 1,\ldots,N-1,\ j = 0,\ldots,N-1, \end{aligned} \tag{7a}$$

$$U_{i,0} = f(ih), \qquad j = 0,\ i = 1,\ldots,N-1, \tag{7b}$$

$$U_{0,j} = p(jk), \qquad i = 0,\ j = 0,1,\ldots,N-1, \tag{7c}$$

$$U_{N,j} = q(jk), \qquad i = N,\ j = 0,1,\ldots,N-1, \tag{7d}$$

where $x_i = ih$, $t_j = jk$ and $r = k/h^2$.

Using equation (7), we obtain the following linear algebraic system of equations:

$$\begin{pmatrix} 2+2r & -r & 0 & 0 & 0 & 0 & 0 \\ -r & 2+2r & -r & 0 & 0 & 0 & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & -r & 2+2r & -r \\ 0 & 0 & 0 & 0 & 0 & -r & 2+2r \end{pmatrix} \begin{pmatrix} U_{1,j+1} \\ U_{2,j+1} \\ . \\ . \\ . \\ U_{N-2,j+1} \\ U_{N-1,j+1} \end{pmatrix}$$

$$= \begin{pmatrix} 2-2r & r & 0 & 0 & 0 & 0 & 0 \\ r & 2-2r & r & 0 & 0 & 0 & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & r & 2-2r & r \\ 0 & 0 & 0 & 0 & 0 & r & 2-2r \end{pmatrix} \begin{pmatrix} U_{1,j} \\ U_{2,j} \\ . \\ . \\ . \\ U_{N-2,j} \\ U_{N-1,j} \end{pmatrix}$$

$$+r\begin{pmatrix} U_{0,j}+U_{0,j+1} \\ 0 \\ . \\ . \\ . \\ 0 \\ U_{N,j}+U_{N,j+1} \end{pmatrix} + 2k \begin{pmatrix} U_{1,j}-{U_{1,j}}^2 \\ U_{2,j}-{U_{2,j}}^2 \\ . \\ . \\ . \\ U_{N-2,j}-{U_{N-2,j}}^2 \\ U_{N-1,j}-U_{N-1,j}^2 \end{pmatrix} \tag{8}$$

Linear system (8) gives $(N-1)$ unknown pivotal values along the boundary $x=1$.
Finally, following discretization is described for the third equation:

$$\begin{aligned} -ra_jU_{i-1,j+1}+(2+2ra_j)U_{i,j+1}-ra_jU_{i+1,j+1} & \\ = ra_jU_{i-1,j}+(2-2ra_j)U_{i,j} & \\ + ra_jU_{i+1,j},\ i=1,\dots,N-1,\ j=0,\dots,N-1, & \quad (9a)\\ U_{i,0}=f(ih), \qquad j=0,\ i=1,\dots,N-1, & \quad (9b)\\ U_{0,j}=p(jk), \qquad i=0,\ j=0,1,\dots,N-1, & \quad (9c)\\ U_{N,j}=q(jk), \qquad i=N,\ j=0,1,\dots,N-1, & \quad (9d)\\ a_j=a(jk), & \quad (9e) \end{aligned}$$

where $x_i=ih$, $t_j=jk$ and $r=k/h^2$.

Using equation (9), we obtain the following linear algebraic system of equations:

$$\begin{pmatrix} 2+2ra_j & -ra_j & 0 & 0 & 0 & 0 & 0 \\ -ra_j & 2+2ra_j & -ra_j & 0 & 0 & 0 & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & -ra_j & 2+2ra_j & -ra_j \\ 0 & 0 & 0 & 0 & 0 & -ra_j & 2+2ra_j \end{pmatrix} \begin{pmatrix} U_{1,j+1} \\ U_{2,j+1} \\ . \\ . \\ . \\ U_{N-2,j+1} \\ U_{N-1,j+1} \end{pmatrix}$$

$$= \begin{pmatrix} 2-2ra_j & ra_j & 0 & 0 & 0 & 0 & 0 \\ ra_j & 2-2ra_j & ra_j & 0 & 0 & 0 & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & ra_j & 2-2ra_j & ra_j \\ 0 & 0 & 0 & 0 & 0 & ra_j & 2-2ra_j \end{pmatrix} \begin{pmatrix} U_{1,j} \\ U_{2,j} \\ . \\ . \\ . \\ U_{N-2,j} \\ U_{N-1,j} \end{pmatrix}$$

$$+ra_j \begin{pmatrix} U_{0,j}+U_{0,j+1} \\ 0 \\ . \\ . \\ . \\ 0 \\ U_{N,j}+U_{N,j+1} \end{pmatrix} \tag{10}$$

Linear system (10) gives $(N-1)$ unknown pivotal values along the boundary $x = 1$.

Problems (2), (3), and (4) can be solved in least-square sense, and a cost function can be defined as a sum of squared differences between measured temperatures and calculated values of $U(x,t)$ by considering guesses estimated values of $q(t)$.

$$f(Guesses \quad estimated \quad values \quad of \quad q(t)) = \sum_{j=1}^{N} (U(a,t_j) - s_j)^2, \tag{11}$$

where $U(a,t_j)$ are calculated by solving the direct parabolic problem. To do this, we consider prior guess for $q(t)$. Also $s_j = s(t_j)$ are measured temperatures at $x = \alpha$. To find optimal solution of $q(t)$, the equation (11) must be minimum.

3 Genetic algorithm (GA) for solving NIPP

Genetic algorithms, primarily developed by Holland [5], have been successfully applied to various optimization problems. It is essentially a searching method based on the Darwinian principles of biological evolution. Genetic algorithm is a stochastic optimization algorithm, which employs a population of chromosomes; each of them represents a possible solution. By applying genetic operators, each successive incremental improvement in a chromosome becomes the basis for the next generation. The process continues until the desired number of generations has been completed or the predefined fitness value has been reached [8].

The genetic algorithms differ from other methods of search and optimization in a number of ways. (a) Genetic algorithms search from a population of possible solutions instead of a single one. (b) The fitness or cost function used to resolve the redundancy has no requirement for continuity in the derivatives; so virtually any fitness function can be selected for optimizing. (c) Genetic algorithms use random operators throughout the process including reproduction, crossover, and mutation. (d) Genetic algorithms are blind, since no specified information about the intended problem is needed to obtain the final solution [8].

In this study a GA is considered for solving IPP. Consider chromosomes $G_i = \{g_{i,1}, g_{i,2}, g_{i,3}, \ldots, g_{i,m}\}$, $i = 1, 2, 3, \ldots, n$ and each $g_{i,j} \in [-M, M]$ (In this work M is one). Each chromosome estimates values of $q(t)$ at $t_j, j = 1, 2, 3, \ldots, n$. Solve then parabolic problem by expressed discretization in previous section. In this work, equation (11) is considered as fitness function than must be minimum. Finally for determining unknown $q(t)$, we find interpolation of m-points of the best chromosome at the end of algorithm. In this study an additional step is added to algorithm after step 4. We named this step repair operator. After applying crossover and mutation, some entries of chromosomes may exceed from $[-M, M]$; so repair operator returns those entries to the interval. The steps of GA for determining $q(t)$ can be divided into the following steps:

1. Generate randomly an initial population of chromosomes.
2. Evaluate the fitness of each chromosome in the population.
3. Choose by tournament selection pairs of chromosome for combination. By applying N-point crossover create offspring of this selected parents.
4. Apply bitwise mutation on offspring.
5. Apply repair operator.
6. Evaluate fitness of offspring.
7. Update population and copy the offspring by α probability.
8. Repeat step 3 to step 7, until finding acceptable fitness.

4 Nelder–Mead simplex search method for solving NIPP

This simplex search method, first proposed by Spendley, Hext, and Himsworth [14] and later refined by Nelder and Mead [9]. Their methods is one of the most efficient pattern search method currently available. This method is a derivative-free line search method that was particularly designed for traditional unconstrained minimization scenarios, such as the problems of nonlinear least squares, nonlinear simultaneous equations, and other types of function minimization [10]. In this method for N vertices of an initial simplex, evaluate cost function for each vertex at the first. Then the worth vertex replace by newly reflected and better point, which can be approximately located in the negative gradient direction. In the minimization problem with three initial simplex vertices, the method can be mention as follows [6, 17]:
x_h: Vertex with highest cost function value.
x_s: Vertex with the second highest cost function value.
x_l: Vertex with lowest cost function value.
x_c: The centroid of vertices except x_h.

1. Reflection. Reflect x_h (Figure1) and find x_0 such that

$$x_0 = 2x_c - x_h.$$

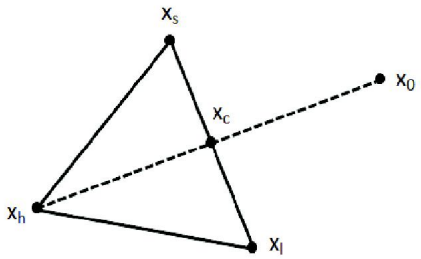


Figure 1: Reflection x_h toward x_0

2. If $f(x_l) < f(x_0) < f(x_s)$, replace x_h by x_0 and return to step 1.

3. Expansion. If $f(x_0) < f(x_l)$, then expansion operation makes x_{00} (Figure2). We replace x_h by x_0 or x_{00} depending on which function value is lower and return to step 1.

$$x_{00} = 2x_0 - x_c.$$

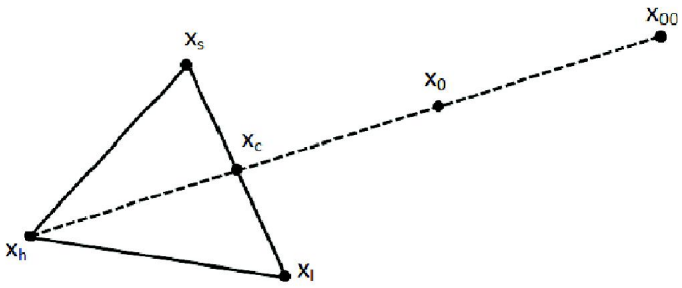


Figure 2: Expansion

4. Contraction. If $f(x_0) > f(x_s)$, then contraction operation makes x_{00} by consider two cases:

(a) If $f(x_0) < f(x_h)$, find x_{00} such that (Figure 3)

$$x_{00} = \frac{1}{2}x_0 - \frac{1}{2}x_c.$$

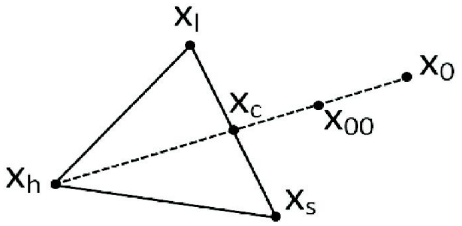


Figure 3: Contraction operator when $f(x_0) < f(x_h)$

(b) If $f(x_0) \geq f(x_h)$, find x_{00} such that (Figure 4)

$$x_{00} = \frac{1}{2}x_h - \frac{1}{2}x_c.$$

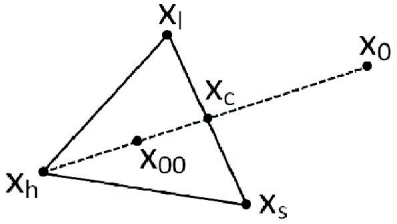


Figure 4: Contraction operator when $f(x_0) \geq f(x_h)$

(c) If $f(x_{00}) < f(x_h)$ and $f(x_{00}) < f(x_0)$, then replace x_h by x_{00} and return to step 1.

(d) If $f(x_{00}) \geq f(x_h)$ or $f(x_{00}) > f(x_0)$, then reduce size of simplex by halving distances from x_l and return to step 1.

The process terminates when either the number of iterations has exceeded a preset amount or the simplex size is smaller than a given value.

In this work, we consider the vectors, which are estimated $q(t)$, as the vertices of the simplex and equation (11) as the cost function. Initial vertices generate randomly. To find unknown $q(t)$, we interpolate the final vertex at the end of NM simplex search method.

5 hybrid algorithm for solving NIPP

GA and NM discussed separately; we now present a hybrid algorithm based on GA and NM. In this algorithm the population size is considered 10. In each iteration we sort the population by fitness of particles. Then top three particles are considered as vertices of NM and lead to NM subroutine, and other seven particles lead to GA subroutine. After applying these subroutines, all of particles are considered as entire of next iteration. The process terminates when pre certain number of iteration done [6]. In this algorithm each particle estimates unknown $q(t)$; also the initial population is generated randomly, and the cost function is considered equation (11). Finally for determining $q(t)$ we find interpolation of m-points of the best particle at the end of algorithm. The steps of hybrid algorithm for determining unknown $q(t)$ can be divided into the following steps:

1. Generate randomly a 10 dimension initial population of particles.
2. Sort population by fitness of particles.
3. Top three particles are lead to NM subroutine.
4. Next seven particles is lead to GA subroutine.
5. Repeat step 2 to step 4, until terminate criteria is not satisfied.

6 Convergence study

The convergence of the Genetic Algorithm and the Nelder–Mead simplex search method have been studied in [13, 15]. Since our presented hybrid method in Section 5 uses the Nelder–Mead simplex search method as a inner subroutine in the Genetic Algorithm, the general convergence is depend on the convergence of the genetic algorithm. Therefore, The hybrid method used in this work converges to optimal solution.

7 Numerical results

In this section we are going to demonstrate numerically, some results for the unknown function $q(t)$ in the NIPPs of Section 2. The aim of this section is to show the applicability of the presented hybrid algorithm for solving NIPPs. To show utility of this algorithm, we solved three examples using GA, NM, and hybrid algorithm separately and compared them with each other. Now, we give the following examples in $0 < x < 1, 0 < t < 1$.

Example 1.

$$U_t(x,t) + U(x,t)U_x(x,t) = U_{xx}(x,t), \quad 0 < x < 1, \quad 0 < t < t_M, \tag{12a}$$
$$U(x,0) = \frac{1}{2} - \frac{1}{2}\tanh(\frac{x}{4}), \quad 0 \le x \le 1, \tag{12b}$$
$$U(0,t) = \frac{1}{2} - \frac{1}{2}\tanh(-\frac{t}{8}), \quad 0 \le t \le t_M,$$
$$U(1,t) = q(t), \quad 0 \le t \le t_M,$$

and the overspecified condition

$$s(t_j) = U(0.5, t_j), \quad t_j = 0.05 \times j, \quad j = 0, 1, 2, \ldots, 20.$$

Here the exact $U(x,t)$ and $q(t)$ are $(\frac{1}{2} - \frac{1}{2}\tanh(\frac{1}{4}(x - \frac{t}{2})))$ and $(\frac{1}{2} - \frac{1}{2}\tanh(\frac{1}{4}(1 - \frac{t}{2})))$, respectively.

Example 2.

$$U_t(x,t) - U(x,t)(1 - U(x,t)) = U_{xx}(x,t), \quad 0 < x < 1, \quad 0 < t < t_M, \tag{13a}$$
$$U(x,0) = \frac{1}{4}(1 - \tanh(\frac{1}{2\sqrt{6}}x))^2, \quad 0 \le x \le 1,$$
$$U(0,t) = \frac{1}{4}(1 - \tanh(-\frac{5t}{24}))^2, \quad 0 \le t \le t_M,$$
$$U(1,t) = q(t), \quad 0 \le t \le t_M,$$

and the overspecified condition

$$s(t_j) = U(0.5, t_j), \quad t_j = 0.05 \times j, \quad j = 0, 1, 2, \ldots, 20.$$

Here the exact $U(x,t)$ and $q(t)$ are $(\frac{1}{4}(1 - \tanh(\frac{1}{2\sqrt{6}}(x - \frac{5t}{2\sqrt{6}})))^2)$ and $(\frac{1}{4}(1 - \tanh(\frac{1}{2\sqrt{6}}(1 - \frac{5t}{2\sqrt{6}})))^2)$, respectively.

Example 3.

$$U_t(x,t) = \frac{2t}{t^2+1}U_{xx}(x,t), \quad 0 < x < 1, \quad 0 < t < t_M, \tag{14a}$$
$$U(x,0) = \frac{1}{3}e^{-x}, \quad 0 \le x \le 1, \tag{14b}$$
$$U(0,t) = \frac{t^2+1}{3}, \quad 0 \le t \le t_M, \tag{14c}$$
$$U(1,t) = q(t), \quad 0 \le t \le t_M, \tag{14d}$$

and the overspecified condition

$$s(t_j) = U(0.9, t_j) + \sigma R, \quad j = 1, 2, 3, \cdots, 9, \tag{14e}$$

where t_j's are the sinc times nodes.

(14f)

Here the exact $U(x,t)$ and $q(t)$ are $e^{-x}(\frac{t^2+1}{3})$ and $e^{-1}(\frac{t^2+1}{3})$, respectively.

Remark 2. In a NIPP there are two sources of error in the estimation. The first source is the unavoidable bias deviation (or deterministic error). The second source of error is the variance due to the amplification of measurement errors (stochastic error). The global effect of deterministic and stochastic errors are considered in terms of the mean squared error or total error, [2].

$$S = \Big[\frac{1}{N-1}\sum_{j=1}^{N}(\widehat{q_j} - q_j)^2\Big]^{\frac{1}{2}}, \tag{15}$$

where N is the total number of estimated values, $\widehat{q_i}$ are calculated values from interpolated equation, and q_i are exact values of $q(t)$. We use from value of total error to compare methods.

7.1 Solving examples by genetic algorithm

In this subsection, at first, we use genetic algorithm for solving examples. Table 1 presents parameters of the proposed genetic algorithm.

In this algorithm, a population of 10 chromosomes of 20 genes is used as the initial guess for numerical results. The gens estimate 20 values of $q(t_j)$ at $t_j = j \times 0.05$, $j = 1, 2, 3, \ldots, 20$. To determine unknown $q(t)$, the best chromosome is interpolated at the end of algorithm. Tables 2–4 present results of implementation of the proposed genetic algorithm to determine unknown $q(t)$ at Examples 1–3, respectively.

Table 1: Parameters of the proposed genetic algorithm

Representation	Real valued vectors
Length of chromosomes	20
Recombination	N point crossover
Recombination probability	100%
Mutation	Swap
Mutation probability	$1/n$
Parent selection	Best of 2 out of random 4
Survivor selection	Replace random
Population size	10
Number of offspring	1
Initialization	Random.
Termination condition	Number of generation

Table 2: The results of 100 to 1000000 generations for determining $q(t)$ at Example 1 by implementing proposed genetic algorithm for a population of 10 chromosomes of 20 genes.

Generation	Best fitness	Time (s)	S
100	0.027745	0.16	0.03165
1000	0.003935	1.01	0.00946
10000	0.000901	5.76	0.00685
100000	0.000097	58.42	0.00279
1000000	0.000051	563.14	0.00047

Table 3: The results of 100 to 1000000 generations for determining $q(t)$ at Example 2 by implementing proposed genetic algorithm for a population of 10 chromosomes of 20 genes.

Generation	Best fitness	Time (s)	S
100	0.014947	0.51	0.29677
1000	0.009230	1.75	0.05083
10000	0.003019	7.19	0.00917
100000	0.000223	63.41	0.00231
1000000	0.000076	688.60	0.00048

Table 4: The results of 100 to 1000000 generations for determining $q(t)$ at Example 3 by implementing proposed genetic algorithm for a population of 10 chromosomes of 20 genes

Generation	Best fitness	Time (s)	S
100	0.010579	0.62	0.01337
1000	0.002323	1.36	0.00915
10000	0.000530	8.78	0.00411
100000	0.000021	59.46	0.00175
1000000	0.000018	480.67	0.00014

7.2 Solving examples by Nelder–Mead simplex search method

To solve presented examples by NM simplex search method, three initial vertices were generated randomly. Each vertex is considered a real valued vector by 20 entries. So each vertex estimates unknown $q(t)$, and each entry of vertices estimates $q(t_j)$ at $t_j = j \times 0.05$,$j = 1, 2, 3, \ldots, 20$. To determine unknown $q(t)$, the best vertex is interpolated at the end of algorithm. Tables 6–7 present results of implementation of NM for determine unknown $q(t)$ at Examples 1–3, respectively.

Table 5: The results of 100 to 1000000 iteration for determining $q(t)$ at Example 1 by implementing NM for three vertices

Iteration	Best fitness	Time (s)	S
100	0.545731	1.07	0.25919
1000	0.526974	7.32	0.24156
10000	0.073689	23.87	0.08513
100000	0.041615	184.09	0.06720
1000000	0.036581	1942.50	0.06669

Table 6: The results of 100 to 1000000 iteration for determining $q(t)$ at Example 2 by implementing NM for three vertices

Iteration	Best fitness	Time (s)	S
100	0.156634	0.89	0.20997
1000	0.147336	2.87	0.13720
10000	0.099725	17.90	0.08711
100000	0.054095	153.17	0.08124
1000000	0.017139	1626.01	0.02814

Table 7: The results of 100 to 1000000 iteration for determining $q(t)$ at Example 3 by implementing NM for three vertices

Iteration	Best fitness	Time (s)	S
100	0.115466	0.73	0.19575
1000	0.109927	1.98	0.12761
10000	0.080606	19.36	0.07540
100000	0.044853	113.71	0.07341
1000000	0.012802	1094.75	0.01937

7.3 Solving examples by hybrid algorithm

In this subsection, Examples 1–3 are solved by proposed hybrid algorithm in Section 5. In this algorithm, a population of 10 vectors of 20 entries is used as the initial guess for numerical results. Therefore, each vector estimates unknown $q(t)$ and each entry of vectors estimates $q(t_j)$ at $t_j = j \times 0.05$,$j = 1, 2, 3, \ldots, 20$. At the each iteration of hybrid algorithm, population is sorted by fitness. Then NM-subroutine is run for 10 iteration on top of three individuals of population and GA-subroutine applies genetic operations (recombination and mutation) on next seven individuals. To determine unknown $q(t)$, the best vertex is interpolated at the end of algorithm. Tables 8–10 present results of implementation of hybrid algorithm for determine unknown $q(t)$ at Examples 1–3, respectively. Figures 5–7 present exact and numeric $q(t)$ for 1000 iterations at Examples 1–3, respectively.

Table 8: The results of 100 to 1000 iteration for determining $q(t)$ at Example 1 by implementing hybrid algorithm for ten vertices

hybrid iterations	Best fitness	Time (s)	S
100	0.000077	49.51	0.00126
1000	0.000039	461.29	0.00041

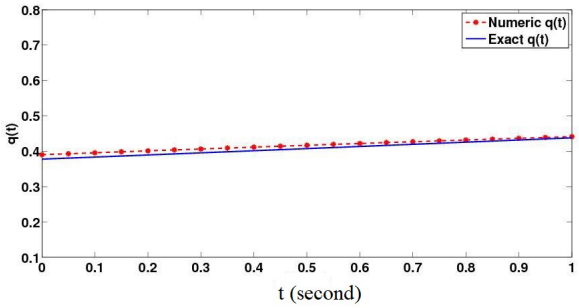


Figure 5: Exact and numeric $q(t)$ for 1000 iterations by implementing hybrid algorithm at Example 1

Table 9: The results of 100 to 1000 iteration for determining $q(t)$ at Example 2 by implementing hybrid algorithm for ten vertices

Hybrid iterations	Best fitness	Time (s)	S
100	0.000105	41.78	0.00212
1000	0.000028	436.22	0.00038

Table 10: The results of 100 to 1000 iteration for determining $q(t)$ at Example 3 by implementing hybrid algorithm for ten vertices

Hybrid iterations	Best fitness	Time (s)	S
100	0.000063	37.19	0.00092
1000	0.000016	389.83	0.00038

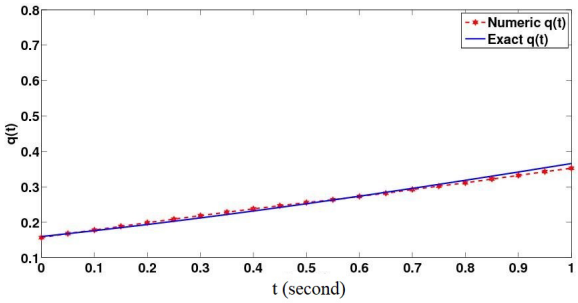


Figure 6: Exact and numeric $q(t)$ for 1000 iterations by implementing hybrid algorithm at Example 2

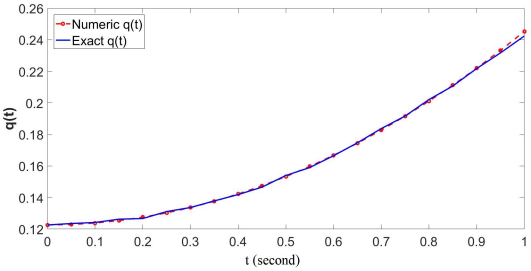


Figure 7: Exact and numeric $q(t)$ for 1000 iterations by implementing hybrid algorithm at Example 3

7.4 Comparison

In this paper, the genetic algorithms, Nelder–Mead simplex search method and hybrid algorithm have explained for solving the nonlinear inverse parabolic problem by some numerical examples. To examine the utility of solutions, we compare the values of total error(s) and their execute times in the estimated solutions. The unknown $q(t)$ has estimated by the some different numbers of iterations for each method. Results show that the accuracy of GA is better than NM for 100 to 1000000 iterations. Figures 8–10 show the convergency study for different iterations for GA and NM at first and second example, respectively. Also execute time of GA is less than NM at both examples in all iterations. But when GA and NM have combined and hybrid algorithm

has created, exploitation of NM and exploration of GA caused that accuracy improve. As total error of hybrid algorithm for 100 and 1000 iterations became better in comparison with the NM and GA performance with almost the same execute time.

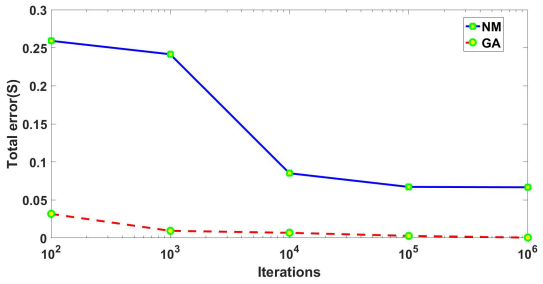


Figure 8: Values of total error(S) for different numbers of iterations by implementing NM and GA at Example 1

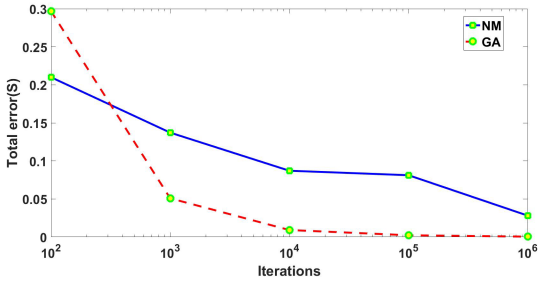


Figure 9: Values of total error(s) for different numbers of iterations by implementing NM and GA at Example 2

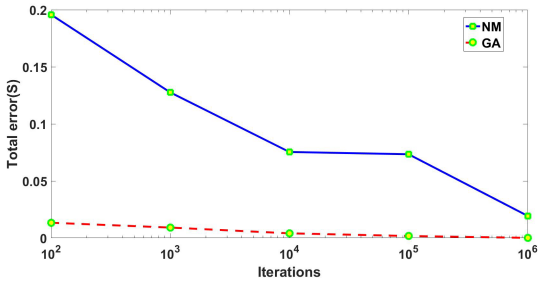


Figure 10: Values of total error(s) for different numbers of iterations by implementing NM and GA at Example3

8 Conclusion

A numerical method to estimate unknown boundary condition is proposed for these kinds of NIPPs, and the following results are obtained:

1. The present study successfully applies the numerical method to NIPPs.
2. To solve the NIPPs by GA, NM, and hybrid algorithm, the unknown function will be guessed and we do not need the regularization. This will improve the execution time.
3. This hybrid algorithm is able to combine whit every direct solution methods.
4. This method does not need to powerful mathematic base.
5. Acceptable accuracy and execute time at the hybrid algorithm.

References

1. Babolian, E. and Saeidian, J. *Analytic approximate solutions to Burgers, Fisher, Huxley equations and two combined forms of these equations*, Commun. Nonlinear Sci. Numer. Simulat., 14 (2009), 1984–1992.
2. Cabeza, J.M.G., Garcia, J.A.M., and Rodriguez, A.C. *A Sequential Algorithm of Inverse Heat Conduction Problems Using Singular Value Decomposition*, Int. J. Thermal Sci., 44 (2005), 235–244.

3. Cantú-Paz, E. *A Summary of Research on Parallel Genetic Algorithms*, IlliGAL Report No 95007, University of Illinois, 1995.

4. Chiwiacowsky, L.D., de Campos Velho, H.F., Preto, A.J., and Stephany, S. *Identifying initial condition in heat conduction transfer by a genetic algorithm: a parallel approach*, in: 24th Iberian Latin American Congress on Computational Methods in Engineering, Ouro Preto, Brazil, 2003.

5. Holland, J.H. *Adaptation in Natural and Artificial System*, University of Michigan Press, Ann Arbor, 1975.

6. Katari, V., Malireddi, S., and Srujan Kollisetty, V.N.K., *An efficient hybrid algorithm for data clustering using improved genetic algorithm and Nelder Mead simplex search*, International Conference on Computational Intelligence and Multimedia Applications, 2007.

7. Levine, D. *A parallel genetic algorithm for the set partitioning problems*, Mathematics and Computer Science Division, Argone National Laboratory Rept., University of Illinois, 1994.

8. Liu, F.-B. *A modified genetic algorithm for solving the inverse heat transfer problem of estimating plan heat source*, Int. J. Heat and Mass Transfer, 51 (2008), 3745–3752.

9. Nelder, J. A. and Mead, R. *A simplex method for function minimization*, Comput. J. 7 (1965), no. 4, 308–313.

10. Olsson, D. M. and Nelson, L. S. *The Nelder–Mead simplex procedure for function minimization*, Technometrics, 7 (1965) 308–313.

11. Pourgholi, R., Tavallaei, N., and Foadian, S. *application of Haar basis method for solving some ill-posed inverse problems*, J. Math. Chem., 8(50) (2012), 2317–2337.

12. Shahrezaee, A. M. and Hoseini nia, M. *Solution of Some Parabolic Inverse Problems by Adomian Decomposition Method*, Appl. Math. Sci., 5(50) (2011), 3949–3958.

13. Sharapov R. R. and Lapshin A. V. *Convergence of Genetic Algorithms*, Pattern Recognition and Image Analysis, 16(3) (2006), 392–397.

14. Spendley, W., Hext, G. R., and Himsworth, F. R. *Sequential application of simplex designs in optimization and evolutionary operation.* Technometrics, 4(4) (1962), 441–461.

15. Mckinnon K. I. M. *Convergence of the Nelder–Mead simplex method to a nonstationary point*, SIAM J. Optim., 9(1) (1998), 148–158.

16. Powell, M.J.D. *On Search Directions for Minimization Algorithms*, Mathematical Programming, 4(2) (1973), 193–201.

17. Walsh, G. R. *An Introduction to Linear Programming*, ISBN 0-471-90719-7 (Wiley), 2nd edition, 1985.

of Translation Persian

Abstracts

حل تقریبی مساله خطی زمان-مکان کسری با استفاده از توابع اندازه‌پذیر

سمانه صردی زید، علی وحیدیان کامیاد، سهراب عفتی

دانشگاه فردوسی مشهد، دانشکده علوم ریاضی، گروه ریاضی کاربردی

دریافت مقاله ۲۱ فروردین ۱۳۹۵، دریافت مقاله اصلاح شده ۲۴ آذر ۱۳۹۵، پذیرش مقاله ۱۳ بهمن ۱۳۹۵

چکیده : در این مقاله، مشتق کسری ریمان-لیوویل را برای توابع انتگرال‌پذیر لبگ توسیع خواهیم داد. سپس این تقریب را برای حل رده‌ای خاص از معادلات دیفرانسیل با مشتقات زمان-مکان کسری به کار خواهیم برد. برای این منظور، مساله فوق را به یک مساله بهینه‌سازی گسسته تبدیل می‌کنیم به گونه‌ای که با یافتن جواب این مساله بهینه‌سازی، تقریب مناسبی از جواب مساله اصلی به دست خواهد آمد. کارآمدی روش را در مثال‌های ارائه شده خواهید دید.

کلمات کلیدی : مشتق ریمان-لیوویل؛ معادله دیفرانسیل کسری؛ معادله دیفرانسیل با مشتقات جزئی کسری؛ توابع انتگرال‌پذیر و اندازه‌پذیر لبگ.

روش شار موج تغییر یافته برای حل معادله اولر یک بعدی با ترمهای منبع ثقلی

حسین مهدیزاده

دانشگاه بیرجند، گروه عمران

دریافت مقاله ۲۳ مهر ۱۳۹۵، دریافت مقاله اصلاح شده ۱۱ بهمن ۱۳۹۶، پذیرش مقاله ۹ اسفند ۱۳۹۶

چکیده : در این مقاله یک روش نوین حجم محدود نوع گودونوف برای حل معادلات اولر یک بعدی ارائه گردیده است. روش عددی معرفی شده در اینجا خوش توازن بوده و قادر به تقریب روش حل با بخش مجموعه ای از نا پیوستگی ها از هر سطح مشترک سلول ریمان می باشد. ترمهای منبع مرتبط نیز در داخل تفاضل شارهای دو سلول محاسباتی روش حجم محدود رفتار می گردند. ابتدا توانایی حل کننده عددی تحت تاثیر ترمهای منبع ثقلی بررسی گردیده و نتایج با حل های مرجع و روش ونو با دقت مرتبه بالا مقایسه می گردد. سپس ویژگی خوش توازن بودن روش مورد نظر برای حل مسائل با حالت ماندگار امتحان گردیده و نهایتا روش پیشنهادی برای مدل سازی اغتشاشات با دامنه کوچک و بزرگ بکار گرفته می شود. نتایج عددی نشان می دهد که روش خوش توازن معرفی شده پیش بینی های دقیقی برای مسائل داده شده ارائه می نماید.

کلمات کلیدی : روش پخش موج؛ روش شار موج؛ حل کننده ریمان؛ خوش توازن؛ معادلات اولر.

روش جستجوی همسایگی متغیر برای مسئله مینیمم رنگ آمیزی مجموع روی گراف‌های ساده

خلیل عرفانی حیدرنیا، صادق رحیمی شعرباف و جعفر فتحعلی

دانشگاه صنعتی شاهرود، گروه ریاضی کاربردی

دریافت مقاله ۱۱ دی ۱۳۹۵، دریافت مقاله اصلاح شده ۲۷ خرداد ۱۳۹۶، پذیرش مقاله ۹ اسفند ۱۳۹۶

چکیده : مسئله رنگ آمیزی مجموع کمینه (MSCP) عبارتست از یافتن رنگ آمیزی راسی مجاز برای گراف G با استفاده از اعداد طبیعی (۱و۲و...) بطوری که مجموع اعدادی که بعنوان رنگ به رئوس نسبت داده شد اند، کمینه شود. هدف اصلی در این مقاله ارایه یک روش جستجوی همسایگی متغیر مورب برای این مسئله، مبتنی بر ساختار های همسایگی جدید است. برای افزایش سرعت جستجوی همسایگی از مفهوم جدیدی بنام نگهدارندگی راسی و مجموعه‌ای استفاده شده‌است. در نهایت عملکرد قابل قبول روش ارایه شده، روی ۲۳ نمونه مسئله متداول، با چند روش ابتکاری در دهه اخیر مورد مقایسه قرار گرفته است.

کلمات کلیدی : مجموع رنگ آمیزی کمینه؛ جستجوی همسایگی متغیر؛ مجموع رنگی؛ راس نگهدارنده؛ مجموعه نگهدارنده؛ مجموعه کاهنده.

بازده به مقیاس تکنولوژیک: شناسایی و ترسیم

انسیه حاجی‌نژاد و محمدرضا علیرضایی

دانشگاه علم و صنعت ایران، دانشکده علوم ریاضی، گروه ریاضی کاربردی

دریافت مقاله ۲۷ آذر ۱۳۹۴، دریافت مقاله اصلاح شده ۱۱ بهمن ۱۳۹۶، پذیرش مقاله ۵ اردیبهشت ۱۳۹۷

چکیده : یکی از مهمترین مسایل برای به کارگیری مدل‌های تحلیل پوششی داده‌ها، شناسایی بازده به مقیاس تکنولوژیک (TRTS) است. اخیرا روش زاویه‌ها مبتنی بر داده‌کاوی برای شناسایی TRTS معرفی شده است. این روش عینی از زاویه‌ها برای اندازه‌گیری شکاف میان TRTS ثابت و متغیر استفاده می‌کند. شکاف در دو بخش افزایشی و کاهشی مرز محاسبه می‌گردد. هر چه شکاف در بخش افزایشی و/یا کاهشی مرز بیشتر باشد، TRTS به شکل افزایشی و/یا کاهشی نزدیک‌تر می‌باشد. در این مقاله، ما روشی ابتکاری برای ترسیم TRTS پیشنهاد می‌دهیم که درک بهتری از TRTS مجموعه داده ارایه می‌دهد. برای این منظور، روش بیشترین زاویه‌ها را برای محاسبه حداکثر انحراف ممکن از فرض TRTS ثابت در بخش‌های افزایشی و کاهشی مرز معرفی نموده‌ایم. با استفاده از زاویه‌ها و بیشترین زاویه‌ها، می‌توان TRTS مجموعه داده را در یک فضای دو بعدی رسم نمود. برای درستیابی روش پیشنهادی، از ۶ نمونه یک ورودی/یک خروجی استفاده نموده‌ایم. همچنین روش زاویه‌ها و روش بیشترین زاویه‌ها را بر روی داده‌های بانک مسکن ایران مورد استفاده قرار داده و TRTS مجموعه داده بانک را در یک فضای دو بعدی رسم کردیم.

کلمات کلیدی : تحلیل پوششی داده‌ها؛ بازده به مقیاس؛ تکنولوژی؛ بانک.

طرح های صریح و ضمنی برای مدل ویروس هانته از مرتبه کسری

مولود یکیت انگان[1] *و داملا ارسلان*[2]

[1] ترکیه، اسپارتا، دانشگاه سلیمان دمیرل، گروه ریاضی
[2] ترکیه، اسپارتا، دانشگاه سلیمان دمیرل، دانشکده علوم طبیعی و کاربردی

دریافت مقاله ۲۷ شهریور ۱۳۹۴، دریافت مقاله اصلاح شده ۲۹ فروردین ۱۳۹۷، پذیرش مقاله ۲۶ اردیبهشت ۱۳۹۷

چکیده :
در این مقاله، یک مدل از مرتبه کسری برای جمعیت موش ها معرفی شده است. برخی از طرح های صریح و ضمنی مانند روش های تتا و تفاضلات متناهی غیر استاندارد (NSFD) برای حل عددی دستگاه معادلات دیفرانسیل معمولی غیر خطی به نام مدل ویروس اپیدمی هانته اجرا شده است. این روش ها با هم مقایسه شده و بحث شده است که این روش خواص مثبت بودن جواب را در حالت سیستم های از مرتبه صحیح را حفظ می کنند. جوا بهای عددی با استفاده از چند نمودار تشریح شده است. نتایج عددی روش های صریح و ضمنی نشان می دهد که این روش ها زمانی که برای مدل ویروس هانته از مرتبه کسری به کار برده می شوند آسان و دارای دقت خوبی هستند.

کلمات کلیدی : روش های صریح و ضمنی؛ روش تتا؛ روش تفاصلات متناهی غیر استاندارد؛ دستگاه معادلات دیفرانسیل غیر خطی از مرتبه کسری؛ مدل جمعیت موش ها.

روش هم محلی گسسته برای معادلات انتگرالی ولترا بطور ضعیف تکین با هسته های لگاریتمی

پیام مختاری

دانشگاه صنعتی سهند تبریز، دانشکده علوم پایه، گروه ریاضی

دریافت مقاله ۱۵ آذر ۱۳۹۵، دریافت مقاله اصلاح شده ۲۶ اردیبهشت ۱۳۹۷، پذیرش مقاله ۲۳ خرداد ۱۳۹۷

چکیده : یک روش هم محلی گسسته مناسب به منظور حل معادلات انتگرالی ولترا بطور ضعیف تکین با هسته های لگاریتمی مورد بررسی قرار گرفته است. یکی از ویژگی های این معادلات این است که در حالت کلی مشتق مرتبه اول جواب مانند یک تابع لگاریتمی رفتار می کند که در مبدا ناپیوسته است. در این مقاله برای ساخت یک جواب تقریبی هم راستا با جواب واقعی یک رویکرد هم محلی جدید را معرفی می کنیم که در آن چند جمله ایهای مونتز–لگاریتمی به عنوان توابع پایه ای بکار برده می شود. بعلاوه، چون پیاده سازی این رویکرد منجر به انتگرال هایی با تکینی های لگاریتمی می شود که اغلب حل آنها به روش عددی مشکل است، یک روش انتگرال گیری عددی مناسب با توابع وزن لگاریتمی را بکار می بریم که درآن انتگرال های چند جمله ای ها با توابع وزن لگاریتمی بطور دقیق محاسبه می شود. بدین منظور روشهای انتگرالگیری گاوس–ژاکوبی را یادآوری نموده و سپس آنرا برای انتگرالهای با توابع وزن لگاریتمی تعمیم می دهیم. آنالیز همگرایی روش پیشنهادی ارائه می شود و برخی نتایج عددی به منظور تایید دقت و مناسب بودن روش پیشنهادی ارائه می شود.

کلمات کلیدی : روش هم محلی گسسته؛ چند جمله ای های مونتز–لگاریتمی؛ روش انتگرالگیری عددی؛ معادلات انتگرالی ولترا بطور ضعیف تکین با هسته های لگاریتمی.

یک الگوریتم ترکیبی کارا بر اساس الگوریتم ژنتیک و روش جستجوی نلدر- مید برای حل مسائل معکوس سهموی غیرخطی

حسن دانا مزرعه و رضا پورقلی

دانشگاه دامغان، دانشکده ریاضی و علوم کامپیوتر

دریافت مقاله ۱۵ اردیبهشت ۱۳۹۶، دریافت مقاله اصلاح شده ۴ آذر ۱۳۹۶، پذیرش مقاله ۲۳ خرداد ۱۳۹۷

چکیده : در این مقاله، یک الگوریتم ترکیبی بر اساس الگوریتم ژنتیک و روش جستجوی سیمپلکس نلدر- مید با روش کمترین مربعات برای تعیین درجه حرارت در مسایل سهموی معکوس غیر خطی ترکیب می‌شود. کارایی الگوریتم ترکیبی با چند مثال از مسائل معکوس غیرخطی سهموی مورد تایید قرار می گیرد. نتایج نشان می‌دهد که این روش ترکیبی، بهتر از الگوریتم ژنتیک و روش جستجوی سیمپلکس نلدر- مید به طور جداگانه است. نتایج عددی با پیاده سازی الگوریتم های مطرح شده در یک پردازنده تک هسته ای با سرعت ۲,۲۰ GHz بدست آمده است.

کلمات کلیدی : ترکیبی؛ NIPP ؛ مسائل معکوس غیرخطی سهموی؛ الگوریتم ژنتیک؛ نلدر-مید؛ روش کمترین مربعات.

Aims and scope

Iranian Journal of Numerical Analysis and Optimization (IJNAO) is published twice a year by the Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad. Papers dealing with different aspects of numerical analysis and optimization, theories and their applications in engineering and industry are considered for publication.

Journal Policy

After receiving an article, the editorial committee will assign referees. Refereeing process can be followed via the web site of the Journal.
The manuscripts are accepted for review with the understanding that the work has not been published and also it is not under consideration for publication by any other journal. All submissions should be accompanied by a written declaration signed by the author(s) that the paper has not been published before and has not been submitted for consideration elsewhere.

Instruction for Authors

The Journal publishes all papers in the fields of numerical analysis and optimization. Articles must be written in English.
All submitted papers will be refereed and the authors may be asked to revise their manuscripts according to the referee's reports. The Editorial Board of the Journal keeps the right to accept or reject the papers for publication.
The papers with more than one authors, should determine the corresponding author. The e-mail address of the corresponding author must appear at the end of the manuscript or as a footnote of the first page.
It is strongly recommended to set up the manuscript by Latex or Tex, using the template provided in the web site of the Journal. Manuscripts should be typed double-spaced with wide margins to provide enough room for editorial remarks.
References should be arranged in alphabetical order by the surname of the first author as examples below:
[1] Stoer, J. and Bulirsch, R. *Introduction to Numerical Analysis*, Springer-verlag, New York, 2002.
[2] Brunner, H. *A survey of recent advances in the numerical treatment of Volterra integral and integro-differential equations*, J. Comput. Appl. Math. 8 (1982), 213-229.

Submission of Manuscripts

Authors may submit their manuscripts by either of the following ways:

a) Online submission (pdf or dvi files) via the website of the Journal at:

http://ijnao.um.ac.ir

b) Via journal's email mjms@um.ac.ir

Copyright Agreement

Upon the acceptance of an article by the Journal, the corresponding author will be asked to sign a "Copyright Transfer Agreement" (see the web site) and send it to the Journal address. This will permit the publisher to publish and distribute the work.

Iranian Journal of Numerical Analysis and Optimization

Former : MRJMS

CONTENTS Vol. 8, No.2, pp 1-140, 2018

web site : http://ijnao.um.ac.ir

Email : ijnao@um.ac.ir

ISSN-Print: : 2423-6977

ISSN-Online: : 2423-6969