# Numerical Computation of Various Fluid Flow Problems

**THESIS**

Submitted to
Babasaheb Bhimrao Ambedkar University
(A Central University)
Lucknow

BABASAHEB
BHIMRAO
AMBEDKAR
UNIVERSITY

LUCKNOW
प्रज्ञा शील करुणा
ESTABLISHED 1996

for the award of the Degree of

**Doctor of Philosophy**

in

**APPLIED MATHEMATICS**


Under the supervision of
**Dr. B.K. SINGH**

Research Scholar
**PRAMOD KUMAR**
Enrollment No. 0927/13

DEPARTMENT OF APPLIED MATHEMATICS
SCHOOL FOR PHYSICAL SCIENCES
BABASAHEB BHIMRAO AMBEDKAR UNIVERSITY
(A CENTRAL UNIVERSITY)
VIDYA VIHAR, RAEBARELI ROAD, LUCKNOW-226 025
UTTAR PRADESH, INDIA


**2018**

# Numerical Computation of Various Fluid Flow Problems

**THESIS**

Submitted to
Babasaheb Bhimrao Ambedkar University
(A Central University)
Lucknow

BABASAHEB
BHIMRAO
AMBEDKAR
UNIVERSITY
LUCKNOW
प्रज्ञा शील करुणा
ESTABLISHED 1996

for the award of the Degree of

**Doctor of Philosophy**

in

**APPLIED MATHEMATICS**


Under the supervision of

**Dr. B.K. SINGH**

Research Scholar

**PRAMOD KUMAR**

Enrollment No. 0927/13

**DEPARTMENT OF APPLIED MATHEMATICS**
**SCHOOL FOR PHYSICAL SCIENCES**
**BABASAHEB BHIMRAO AMBEDKAR UNIVERSITY**
**(A CENTRAL UNIVERSITY)**
**VIDYA VIHAR, RAEBARELI ROAD, LUCKNOW-226 025**
**UTTAR PRADESH, INDIA**


**2018**

**Dedicated**

**to**

**My Family Members**

# DECLARATION

I declare that the work embodied in this Ph.D. thesis entitled as **"Numerical Computation of Various Fluid Flow Problems"** is carried out by me under the supervision of Dr. B.K. Singh, Department of Applied Mathematics, Babasaheb Bhimrao Ambedkar University (A Central University) Lucknow, India. The information presented in this Ph.D. thesis has not been submitted for the award of any degree or diploma. I declare that I have faithfully acknowledged, given credit to and referred to the research workers whenever their works have been cited in the text and the body of the thesis. I further declare that I have not willful lifted up some others work, para, text, data, results, etc. reported in the journals, books, magazines, reports, dissertations, thesis, etc., or available at websites and included them in this thesis and cited my own work.

Date: 28-09-2018

Pramod Kumar

Pramod Kumar

(Research Scholar)

# CERTIFICATE

This is to certify that the thesis titled "**Numerical Computation of Various Fluid Flow Problems**" submitted by **Mr. Pramod Kumar** is an original research work and has not been previously submitted in part or full for the award of any other degree or diploma to this or any other university.

The thesis submitted to the Babasaheb Bhimrao Ambedkar University Lucknow satisfies all the requirements as stipulated in the *Doctor of Philosophy (Ph.D.) regulations -1999 as amended in 2010* and it is fit for submission and evaluation for the award of the degree of Doctor of Philosophy of the University.

Date: 28-09-2018

Supervisor

Dr. Brajesh Kumar Singh
Assistant Prof.
Department of Applied Mathematics
B. B. Ambedkar University Lucknow

Head of the Department

28.09.2018

Professor B. S. Bhadauria
Head
Department of Mathematics
B.B. Ambedkar University Lucknow

# Acknowledgement

A positive energy flowing within me that has been a source of motivation, confidence and courage at every walk of my life, do not permit me to undermine the existence of supreme authority of *GOD*! without his divine intervention the completion of my research work has not been possible.
I feel its true that "*If you strongly desire something with your heart, the whole world conspires to fulfill your desire*".

With great pleasure, I take this opportunity to thank all those whose advice and encouragement have been received. At this important stage of my life, they all deserve a recognition.

It is a great pleasure for me to express my sincere appreciation to my supervisor Assistant Professor Dr. Brajesh Kumar Singh for guiding and motivating me throughout the work. I would like to thank him not only for helping me to overcome some obstacles along the path but also for contributions of him to my life.

I would like to express my gratitude to Prof. B.S. Bhadauria (Head, Department of Mathematics, BBA University Lucknow), all the Research committee, Department Research committee and faculty members for encouraging me and providing the necessary facilities for carrying out the research work. I wish to thank my friends Manoj K. Singh, Vineet Kumar, Ajay Singh, Kanchan Shakya and Neetu Singh for their support. Thanks also go out to each scholar, non-teaching staff in the Department of Mathematics and well wishers for their any direct or indirect support during my research work. I extended my warm and sincere thanks to Assistant Professor Manoj Kumar (DCS, BBAU, Lucknow) for his valuable advise at any time any kind help and support. The support of all is highly appreciated.

The financial assistance from the Babasaheb Bhimrao Ambedkar University, Lucknow.

My family is my fortress! The most special thanks goes to my parents: Shri Bhagwati Prasad and Smt. Raj Kumari for their love, kind affection, moral support and never ending blessing. I express my heartfelt thanks and deepest regards to my sisters Kiran, Neera, Dauli and Mamta for her understanding, kind affection, encouraging thoughts and moral support which help me to carry out the research work smoothly. My brother: Bhoopendra K. Singh is very nice, Sincere and always carrying to me. I am short of words, when it comes to express my heartfelt appreciation for what all my family do. They have been the greatest source of inspiration behind my all the achievements.

With devotion and profound gratitude, I dedicated this thesis to my *parents*.

Lucknow (Pramod Kumar)

# Abstract

Partial differential equations (PDEs) are the basic tools of various mathematical models occurred not only in physical, chemical and biological phenomena but also in the other fields such as in economics, financial forecasting, image processing and many more. Notice that physical phenomena consisting of certain hereditary properties can't be explained via a mathematical model involving only classical differential/integral operators as they are local in nature. The fractional differential/integral operators, introduced by a great mathematician Leibniz (1965), are non-local in nature (i.e., the next state of a system depends not only upon its current state but also upon all of its previous states) and have memory effects as well as an embedded capability to explain the physical phenomena which are not explained accurately in terms of the classical operators, and so, the fractional operators are more realistic and become very popular in modeling certain complex systems arising in fluid mechanics, viscoelasticity, mathematical biology, life sciences, electrochemistry, physics, economics, control theory and many more. The above finding invoked the researchers to develop the techniques to study the behavior of such type of the PDEs model. In the past years, various methods have been developed to study these models of PDEs, among them, finite difference method, compact finite difference method (Lele, 1992), differential quadrature method (Shu, 2000) and collocation methods are employed very broadly to solve classical PDEs.

The main aim of this thesis is to present a numerical study of some fluid flow problems occurred in the form of time-dependent partial differential equations (PDEs). We consider both classical and fractional model of PDEs. Specially, in classical order PDEs, we consider Burgers' equations in one and two dimensions and *Kuramoto-Sivashinsky* equation in one dimension whereas in a fractional model of PDE, we consider coupled viscous Burgers

equations, Navier-Stokes equations and fractional model of PDEs with proportional delay. These classical model of PDEs have been studied numerically using Bellman's differential quadrature method (DQM) (Bellman et al, 1975) with a different set of base functions and Lele's compact finite difference schemes (Lele, 1992). The fractional model of PDEs have been studied by using a fractional variational technique, reduced differential transform method and perturbation techniques.

**Chapter 1** deals with introductory information on the work done. Besides some basic definitions and preliminaries, used throughout the work, we present a brief introduction to differential quadrature method, reduced differential transform method and perturbation techniques and their existing literature review.

In **Chapter 2**, a novel approach: *modified extended cubic B-spline differential quadrature (mECDQ) method* in space discretization has been developed with time integration algorithm for numerical simulation of initial values system of nonlinear Burgers' equation in $(1+1)$ dimension and coupled Burgers equation in $(n+1)$ dimension $(n = 1, 2)$ with appropriate Boundary conditions. The mECDQ method, DQM with modified extended cubic B-splines as base functions, is used to convert the initial boundary value system of the Burgers' equation into an initial value system of ordinary differential equations (ODEs), in time. We prefer an optimal five stage four order strong stability preserving Runge-Kutta method (SSP-RK54) to solve this resulting system of ODEs. Six test problems are considered to test the accuracy and efficiency of mECDQ method. The proposed results are compared with the exact solutions in terms of $L_2$ and $L_\infty$ errors and the existing results. The mECDQ scheme is shown conditionally stable Burgers' equations.

The mECDQ method, DQM with modified extended cubic B-splines as base functions, is used to convert the initial boundary value system of the Burgers' equation into an initial value system of ODEs, in time. We prefer an optimal five stage four order strong stability preserving Runge-Kutta method (SSP-RK54) to solve this resulting system of ODEs. Six test problems are considered to test the accuracy and efficiency of mECDQ method. The proposed results are compared with the exact solutions in terms of $L_2$ and $L_\infty$ errors and the existing results. The mECDQ scheme is shown conditionally stable Burgers' equations.

**Chapter 3** concerns with a new method *modified trigonometric cubic B-spline differential quadrature method (MTB-DQM)* in space discretization with SSP-RK54 algorithm for solving the time-dependent PDEs. Specially, the proposed algorithm has been implemented for nonlinear Burgers' equations. First, MTB-DQM (DQM with modified trigonometric cubic B-splines as base functions) is used to convert the initial boundary value system of Burgers' equation into an initial value system of first order ODEs, in time, thereafter SSP-RK54 algorithm has been employed for solving the resulting system of ODEs. Four test problems are considered to illustrate the accuracy/efficiency of the method in terms of $L_2$ and $L_\infty$ error norms and their comparisons with existing results. Moreover, MTB-DQM is shown conditionally stable for various grid points and computed results are better than the results obtained by almost all the existing schemes.

**Chapter 4** adopted a *compact finite difference scheme* (Lele, 1992) in the space discretizations while optimal four-stage, order three strong stability-preserving time-stepping Runge-Kutta scheme in the time, for the numerical simulation of one dimensional the Kuramoto-Sivashinsky equation "$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \beta\frac{\partial^2 u}{\partial x^2} + \gamma\frac{\partial^4 u}{\partial x^4}$", arises in the study of flame front propagation, phase turbulence in a reaction-diffusion system and many other biological and chemical processes. The efficiency of the proposed scheme confirmed by six test problems with known exact solutions. The numerical results demonstrate the reliability and efficiency of the algorithm developed.

**Chapter 5** deals with an analytical study of time-fractional Navier-Stokes equation, considering the fractional derivative of Caputo type:

$$\mathcal{D}_t^\alpha U + (U \cdot \nabla)U = \nu\nabla^2 U - \tfrac{1}{\rho}\nabla p, \quad \text{on } \Omega \times (0, T)$$

The approximate analytical solutions are obtained by adopting two reliable methods: Fractional reduced differential transform method and a new integral projected differential transform method. The accuracy/efficiency of these methods is illustrated by three test problems of the time fractional Navier-Stokes equation. The scheme is found to be very reliable, effective and efficient powerful technique to solve a wide range of problems arising in engineering and sciences. The small size of computation FRDTM contrary to the other schemes is its

strength.

**Chapter 6** deals with an approximate analytical solution of multi-dimensional, time-fractional coupled viscous Burgers' (TFCB) equation obtained by employing "homotopy perturbation method". The validity and efficiency of the homotopy perturbation method have been illustrated by considering three different examples of TFCB equation. The results are also depicted in graphically for different values of fractional order $\alpha$ and Reynolds number. It is found that the proposed series solutions converge rapidly for large Reynolds numbers (Re$\geq$ 100).

In **Chapter 7**, at first some properties of $(n+1)$-dimensional Extended FRDTM for delayed TFPDEs are presented. Approximate analytic solutions of $(1+1)$ dimensional TF-PDEs with proportional delay and generalized Burgers' equations with proportional delay are obtained by two reliable methods: 1) fractional variation iteration method (FVIM), and 2) Extended fractional reduced differential transform method (Extended FRDTM). The approximate solutions from either method are obtained in a series form that converges to the exact solution behaviors very fast. The efficiency and validity of these methods are illustrated by three test problems of TFPDEs with proportional delay. The finding shows that Extended FRDTM is easy to implement as compared to FVIM. The small size of computation of Extended FRDTM is its strength.

In **Chapter 8**, the *homotopy perturbation transform method* (HPTM) (i.e., hybrid of homotopy perturbation technique & Laplace transform) has been implemented for solving initial value autonomous system of time-fractional partial differential equations (TFPDEs) with proportional delay, including generalized Burgers' equations with proportional delay. The numerical study of three examples of TFPDEs with proportional delay are presented to test the efficiency and validity of proposed HPTM. The obtained solutions are in series form, converges very fast. The HPTM seems very reliable, effective and efficient powerful technique for study of many physical models arising in various branches of sciences and engineering.

# List of Publications

**Journals**

1. Singh BK, **Kumar P**, A novel approach for numerical computation of Burgers' equation in $(1 + 1)$ and $(2 + 1)$ dimensions, *Alexandria Engineering Journal*(2016) 55, 3331–3344, Elsevier. (Chapter 2)

2. Singh BK, **Kumar P**, An algorithm based on DQM with modified trigonometric cubic B-Splines for solving coupled viscous Burgers' equations, *Communications in Numerical Analysis (2017)*, http://dx.doi.org/10.5899/2017/cna-00333. (Chapter 3)

3. Singh BK, Arora G, **Kumar P**, A note on solving the fourth-order Kuramoto-Sivashinsky equation by the compact finite difference scheme. *Ain Shams Engineering Journal*(2016). http://dx.doi.org/10.1016/j.asej.2016.11.008, Elsevier. (Chapter 4)

4. Singh BK, **Kumar P**, FRDTM for numerical simulation of multi-dimensional, time-fractional model of Navier-Stokes equation, *Ain Shams Engineering Journal* (2016). http://dx.doi.org/10.1016/j.asej.2016.04.009, Elsevier. (a part of Chapter 5)

5. Singh BK, **Kumar P**, Kumar V, Homotopy perturbation method for solving time fractional coupled viscous Burger equation in $(2 + 1)$ and $(3 + 1)$ dimensions, *Int. J. Appl. Comput. Math.*, https://doi.org/10.1007/s40819-017-0469-3. (Chapter 6)

6. Singh BK, **Kumar P**, Fractional variational iteration method for solving fractional partial differential equations with proportional delay. *Int. J. of Diff. Eq.* 2017 (2017) Article ID 5206380, 11 pages, Hindawi. (a part of Chapter 7)

7. Singh BK, **Kumar P**, Extended fractional reduced differential transform for solving fractional partial differential equations with proportional delay, *Int. J. Appl. Comput. Math.*, https://doi.org/10.1007/s40819-017-0374-9, Springer. (a part of Chapter 7)

8. Singh BK, **Kumar P**, Homotopy perturbation transform method for solving fractional partial differential equations with proportional delay. *SeMA Journal*, https://doi.org/ 10.1007/s40324-017-0117-1(available at https://arxiv.org/abs/1611.06488v1). Springer. (Chapter 8)

9. Singh BK, **Kumar P**, An algorithm based on a new DQM with modified extended cubic B-splines for numerical study of two dimensional hyperbolic telegraph equation, *Alexandria Engineering Journal*, https://doi.org/10.1016/j.aej.2016.11.009, Elsevier. (**Not a part of this Thesis**)

10. Singh BK, **Kumar P**, An algorithm based on a new DQM with modified exponential cubic B-splines for solving hyperbolic telegraph equation in (2+1) dimension, Nonlinear Engineering (2017), DE Gruyter. https://doi.org/10.1515/nleng-2017-0106. (**Not a part of this Thesis**)

**Communicated**

1. Singh BK, **Kumar P**, A new integral projected differential transform method for time-fractional Navier-Stokes equation. *communicated*. (a part of Chapter 5)

**Papers presented in conferences**

**International**

1. Singh BK, **Kumar P**, A numerical simulation on one dimensional coupled viscous Burger equation using modified cubic B- spline differential quadrature method, presented in International Conference on *Modern Mathematical Methods and High Performance Computing in Science and Technology*, RKGIT, Ghaziabad December 27-29, 2015.

2. Singh BK, **Kumar P**, A numeric study of time fractional model of multi-dimensional coupled viscous Burgers equation, presented in International Conference on *Fluid Dynamics and its applications*, BNMIT, Bengaluru. July 12-14, 2017.

**National**

1. Singh BK, **Kumar P**, Modified cubic b-spline differential quadrature method on Fisher reaction diffusion equation. National conference on *Science For Society: An Interdisciplinary Approach*, BBAU, Lucknow. October 31- November 2, 2015.

2. Singh BK, **Kumar P**, A novel approach for numerical solution of coupled viscous burger equation. National conference on *Mathematical Techniques in Engineering and Technology*, BBAU, Lucknow. March 30-31, 2016.

3. Singh BK, **Kumar P**, A new integral projected DTM for solving fractional model of Navier-Stokes equation, in the unsteady flow of a viscous fluid. National conference on *Science Technology & Innovation for Sustainable Development*, BBAU, Lucknow. March 3-4, 2017.

**Conferences/Workshop Attended**

1. International conference on *Emerging Trends in Computational and Applied Mathematics*, ITM University, Gurgaon. June 2-4, 2014.

2. National conference on *Recent Advances in Mathematics and Applications*, BBAU, Lucknow. October 30-31, 2014.

3. International conference on *Modelling and Computing*, BBAU, Lucknow. July 10-11, 2014.

4. **One week workshop** on *Emerging Research Trends in Computer Science*, BBAU, Lucknow. March 20-24, 2017

# List of Tables

# Contents

# Chapter 1

# Introduction

First, some preliminaries, mathematical methods, literature review and applications of the considered problems are revisited to comprehend the upcoming chapters

## 1.1 Basic definitions and notations

A **fluid [163]** is a substance that has no fixed shape and it deforms continuously when subjected to a shear stress, no matter how small. Fluid is the subset of matter, e.g. liquids, gases, plasmas and some plastic solids.

**Definition 1.1.1** (Density)**.** *The density* ($\rho$) *of a fluid material is its mass* ($m$) *per unit volume* ($V$)*, that is,* $\rho = \frac{m}{V}$*. The SI unit of density is* $kg/m^3$*.*

The principle of density given by Archimedes. The density of the fluid is temperature/pressure dependent, that is, density decreases with increasing temperature while it increases as pressure increases.

**Definition 1.1.2** (Fluid pressure)**.** *Fluid pressure is the force exerted by the fluid per unit area. Fluid pressure is transmitted with equal intensity in all directions and acts normal to any plane. In the same horizontal plane, the pressure intensities in a liquid are equal.*

**Definition 1.1.3** (Equation of Continuity: Conservation of mass)**.** *In physics, the equation of continuity describes the transport of some quantity in a flow. The equation of continuity*

*states that* 'fluid flow in such a way that mass is conserved*'. Mathematically equation of continuity in steady state is*

$$\frac{\partial \rho}{\partial t} + \nabla.\boldsymbol{q} = 0 \tag{1.1.1}$$

where $\rho \rightarrow$ fluid density, $t \rightarrow$time, $\mathbf{q} \rightarrow$ flow velocity. In particular, for incompressible fluid flow, the equation of continuity states that the divergence of the velocity field is zero, that is, $\nabla.\mathbf{q} = 0$.

**Definition 1.1.4** (Classical Navier-stokes Equation: Conservation of momentum)**.** *The fundamental governing equation for incompressible non-newtonian fluid flow, the Navier-Stokes equation is one of the most powerful model was derived in 1822 [180].*

*The general classical Navier-stokes Equation [25] is given by*

$$\frac{\partial \boldsymbol{q}}{\partial t} + \boldsymbol{q}.\nabla \boldsymbol{q} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \boldsymbol{q} + F \tag{1.1.2}$$

*where $\nu \rightarrow$kinematic viscosity, $F \rightarrow$body force (an external forces that act on the fluid, for example: gravity, wind, etc). The local time derivative term $\frac{\partial}{\partial t}$ denote the time rate of change at a fixed point, $\nu\nabla^2\boldsymbol{q}$ viscosity term, $-\frac{1}{\rho}\nabla p$ be the pressure term, $\boldsymbol{q}.\nabla\boldsymbol{q} \rightarrow$mass of fluid, $\boldsymbol{q}.\nabla$ is the convective derivative.*

**Definition 1.1.5** (Viscosity)**.** *The fluid property 'viscosity' is the measure of its resistance, i.e., the intermolecular force between fluid particles to gradual deformation by shear stress. It contradicts the relative motion between the two layers of the fluid.*

Mainly the viscosity is of two type, namely i) *Dynamic viscosity*; *ii*) *Kinematic viscosity.* i) **Dynamic viscosity** sometimes known as shear viscosity is the tangential force per unit area required to one or more horizontal planes with respect to another plane. Dynamic viscosity can be expressed as

$$\tau = \mu \frac{dv}{dy} \tag{1.1.3}$$

which is also called Newton's law of friction, where $\tau \rightarrow$ shearing stress, $\mu \rightarrow$dynamic viscosity, $v \rightarrow$ velocity, $dy \rightarrow$ distance between layers.

ii) **Kinematic viscosity** ($\nu$) is the dynamic viscosity $\mu$ of a fluid per unit density $\rho$, that is, $\nu = \frac{\mu}{\rho}$.

The dynamic viscosity is involved in the relationship between stress and strain tensors. The kinematic viscosity mainly is involved in the classical Navier - Stokes equation governing the fluid motion. If one interested in the interaction between molecules that can be interpreted in terms of mechanical stress, the dynamic viscosity is more appropriate. Nevertheless, the kinematic viscosity is recommended when one interested in fluid motion and velocity field.

**Definition 1.1.6** ( [166, 194]). *A real valued function $\phi$ defined on $\mathbb{R}^+$ belongs to $\mathbb{C}_\nu$ class ($\nu \in \mathbb{R}$) whenever $\exists r(r > \nu) \in \mathbb{R}$ and a function $h \in C[0, \infty)$ such that $\phi(x) = h(x)x^r$ for all $x \in \mathbb{R}^+$. Further the function $\phi$ belongs to the class $\mathbb{C}_\mu^n$, $n \in \mathbb{N}$ whenever nth derivative of $\phi$ in the member of $\mathbb{C}_\nu$.*

In the literature many types of definitions of fractional differentiation/integration operators (e.g., Riemann-Liouville [166], Caputo type [39], He [17], Atangana and Baleanu [17], He's fractional derivatives [82] and many more) has been defined in [191]. Among them, we revisit the basic definitions and preliminaries of due to Riemann-Liouville and Caputo

**Definition 1.1.7** (Riemann-Liouville (R-L) fractional integration [166, 194]). *Let $f \in \mathbb{C}_\mu$, then R-L fractional integral operator $\mathcal{J}_t^\alpha(f)$ ($\alpha \geq 0$) of $f$ is defined by $\mathcal{J}_t^0 f(t) = f(t)$ and*

$$\mathcal{J}_t^\alpha f(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t - \tau)^{\alpha-1} f(\tau)\, d\tau, \quad \text{if } \alpha > 0, \tag{1.1.4}$$

*where*

$$\Gamma(t) = \int_0^\infty e^{-x} x^{t-1} dx, t \in \mathbb{C} \tag{1.1.5}$$

*is the well known gamma function, a generalization of factorial function. In particular $\Gamma(n+1) = n!$ whenever $n \in \mathbb{N}$.*

If $\phi \in \mathbb{C}_\nu, \nu \geq -1, \sigma, \delta \geq 0$ and $\gamma > -1$, then the operator $\mathcal{J}_t^\sigma$ satisfy the following properties

$$a)\ J_t^\sigma J_t^\delta \phi(t) = J_t^\delta J_t^\sigma \phi(t), \qquad b)\ J_t^\sigma t^\gamma = \frac{\Gamma(1+\gamma)}{\Gamma(1+\gamma+\sigma)} t^{\sigma+\gamma}.$$

Riemann - Liouville derivative exists for every continuous function but the Riemann - Liouville derivative of constant is nonzero. Caputo and Mainardi [39] defined a fractional differentiation operator $D_t^\alpha$ to describe the theory of viscoelasticity to overcome the discrepancy of R-L derivative.

**Definition 1.1.8** (Caputo Fractional differentiation [39, 166, 194]). *Let $\phi \in \mathbb{C}_\nu^m,\ \nu \geq -1$. The Caputo fractional derivative $\mathcal{D}_t^\alpha \{f(t)\}$ of $f \in \mathbb{C}_\nu$ of order $\alpha$ $(m-1 < \alpha \leq m \in \mathbb{N})$ is defined by*

$$\mathcal{D}_t^\alpha \phi(t) = \mathcal{J}_t^{m-\alpha} \mathcal{D}_t^m \phi(t), \quad \text{for } t > 0. \tag{1.1.6}$$

Some properties of the differential operator $\mathcal{D}_t^\alpha$ follows the following axioms

**Lemma 1.1.9** ( [39, 194]). *Let $m-1 < \alpha \leq m, m \in \mathbb{N}$, and $\phi \in \mathbb{C}_\nu^m, \nu \geq -1$, $\gamma > \alpha - 1$, and let $C$ be a constant, then*

$$\begin{aligned}
&(a)\ \mathcal{D}_x^\alpha x^\gamma = \frac{\Gamma(1+\gamma)}{\Gamma(1+\gamma-\alpha)} x^{\gamma-\alpha};\\
&(b)\ \mathcal{D}_x^\alpha C = 0;\\
&(c)\ \mathcal{D}_x^\alpha \mathcal{J}_x^\alpha \phi(x) = \phi(x);\\
&(d)\ \mathcal{J}_x^\alpha \mathcal{D}_x^\alpha \phi(x) = \phi(x) - \sum_{k=0}^{m} \phi^{(k)}\left(0^+\right) \frac{x^k}{k!}, \quad \text{for } \mathrm{x} > 0.
\end{aligned} \tag{1.1.7}$$

Notice that the Caputo fractional derivative deals with traditional initial and boundary conditions in the formulation of the physical problems. The readers are suggested to see [22, 40, 70, 85, 86, 96, 120] for more details on fractional calculus.

**Example 1.1.10.** *Let $\alpha = 1/2, n = 1$ and $f(t) = t$, then formula (1.1.6) yields*

$$D_t^{1/2} t = \frac{1}{\Gamma(1/2)} \int_0^t \frac{1}{(t-\tau)^{1/2}} d\tau = \frac{2\sqrt{t}}{\sqrt{\pi}}.$$

*The Caputo fractional derivative of function $f(t) = t$ is obtained as follows*

$D_t^{1/2} t = -\frac{1}{\sqrt{\pi}} \int_0^t \frac{1}{(t-\tau)^{1/2}} d(t-\tau) = -\frac{1}{\sqrt{\pi}} \int_{\sqrt{t}}^0 \frac{dx^2}{x} = \frac{2}{\sqrt{\pi}} \int_0^{\sqrt{t}} dx = \frac{2\sqrt{t}}{\sqrt{\pi}}$.

*which can be obtained directly from Lemma 1.1.9(b) with $\alpha = 1/2, \gamma = 1$.*

## 1.2 Motivation

Partial differential equations are the basic tools of not only in many mathematical models occurred in physical, chemical and biological phenomena but also in the other fields like in economics, financial forecasting, image processing and many more.

The classical (integer order) differential/integral operators being local, it is not possible to construct a mathematical model for each physical phenomenon in terms of classical differential/integral operators. Specially, a physical phenomena consisting of specific hereditary properties can't be explained via a mathematical model involving only classical differential/integral operators. The concept of the fractional order differential operator was given by a great mathematician Leibniz in a letter to L'Hospital, in 1965. The fractional order differential/integral operator is non-local (i.e., the next state of a system depends not only upon its current state but also upon all of its previous states) and have memory effects as well as embedded capability to explain the physical phenomena which are not explained accurately in terms of classical differential operators. Due to the nonlocal property, the fractional operators are more realistic and become very popular in modeling certain complex systems arising in fluid mechanics, viscoelasticity, mathematical biology, life sciences, electrochemistry, physics, economics, control theory, biophysics [22,40,70,84–86,96,121,166,194] and many more. Especially in nano-hydrodynamics where continuum assumption does not well, and the fractional model can be considered to be a best candidate.

## Solution of time dependent PDEs

The computation of exact or approximate solutions of nonlinear PDEs in the various fields of sciences and engineering is still a very significant problem that needs new methods to predict the exact or approximate solution behavior. The above finding invoked the researchers to develop the techniques to investigate the behavior of the PDEs model involving integer order differential/integral operators (and also the PDEs models involving the fractional differential/integral operators). The behavior of the PDEs model can be predicted by adopting analytical as well as numerical techniques.

## 1.3 Analytical techniques

The prediction of the exact solution behavior of classical/fractional differential equation is very tough task. Indeed, various type of vigorous techniques have been developed, in the past years, for computing an approximate solution behavior of such type of differential equations, among them, the homotopy analysis method [208], homotopy perturbation method [86], modified Laplace decomposition method [136], generalized DTM [155], local fractional variational iteration method [283], decomposition method [8, 177], variational iteration method (VIM) [6, 48, 63, 92, 93, 232, 280] and many more. But most of these methods have their inbuilt deficiencies like calculation in the computation of Adomian's polynomials, Lagrange multiplier, divergent results, and huge computational task. A brief description of some of these analytical methods is as follows

### 1.3.1 Variation iteration method

The variation iteration method (VIM) was described by He [83] to solve nonlinear PDEs without linearization or small perturbations. In this approach, a correction functional formula is constructed via Lagrange multiplier and variational theory. The fractional VIM was developed by Odibat and Momani [186]. After the seminal work by He, and Odibat and Momani, the VIM has been used widely to solve not only classical differential equations but also the fractional PDEs, see [6, 48, 63, 71, 85, 92, 93, 188, 189, 207, 232, 280, 283]. The detailed description of variation iteration method is given in Chapter 7.

### 1.3.2 Perturbation methods

In the last four decades with a rapid development of nonlinear science create an interest among researchers to develop analytical techniques for studying the nonlinear differential equations in the various fields of science and engineering. One of the well-known techniques widely applied is the perturbation technique. The main drawback of the perturbation technique is that it is limited to the study of the nonlinear problems in which a small parameter must exist with the nonlinear term of the problems. This small parameter

assumption greatly restricts applications of perturbation technique because it can not apply to a variety of nonlinear problems having no small parameter at all.

### A. Homotopy Perturbation method

In 1999, He [86–90] developed the homotopy perturbation method (HPM) by merging the standard homotopy with perturbation for solving various physical problems. It is worth mentioning that the HPM is employed without any discretization, restrictive assumption or transformation. Contrary to the general perturbation technique, the presence of a small parameter is not required in the differential equations. The HPM in topology is adopted to construct a standard homotopy for the problem by embedding a parameter $p \in [0, 1]$, which is assumed as a "small parameter". The approximate results obtained by employing are uniformly valid/converges not only for problems with small parameters but also for the problems with large parameters [20, 33]. After the seminal work of He, HPM has been widely for the study of a variety of vigorous linear and nonlinear problems of classical and fractional partial differential equations in sciences and engineering, see [20, 32, 33, 67–69, 79, 91, 103, 106, 175, 178, 187, 199, 206, 229, 281] and many more. The detailed description of HPM for classical and fractional partial differential equations is reported in Chapter 6.

### B. Homotopy perturbation transform method

The Laplace transform is incapable of handling nonlinear equations due to the difficulties caused by the nonlinear terms. This difficulty is resolved by developing some methods such as Adomian decomposition method (ADM) [177], Laplace decomposition algorithm [145, 293,294], variational iteration method [83], homotopy analysis method [88,119,137,200,208].

The homotopy perturbation transform method is a hybrid method, which is obtained by merging standard homotopy perturbation method with well known Laplace transform [159]. The advantage of HPTM is its capability of merging two powerful methods for computing the exact solutions for nonlinear equations. The HPTM provides the solution in a rapidly convergent series. This method is a highly effective technique for handling a broad class of nonlinear problems and makes the solution procedure highly convenient as compared to

HPM and ADM. The detailed description of the method is presented in Chapter 8.

### 1.3.3 Differential transform method

The concept of semi-analytical technique "differential transform method (DTM)" is proposed first by Zhou in 1986 [301] for solving linear as well as nonlinear initial value problems in electrical circuit analysis.

#### A. One and two-dimensional differential transform method

The differential transform of $k$th derivative $\phi^{(k)}(\cdot)$ of a function $\phi$ is defined by

$$\Phi(k) = \frac{\phi^{(k)}(x_0)}{k!} \tag{1.3.1}$$

and inverse differential transform of $\Phi$'s is as follows

$$\phi(x) = \sum_{k=0}^{\infty} \Phi(k)(x - x_0)^k \tag{1.3.2}$$

Equation (1.3.1) and (1.3.2) yields the following

$$\phi(x) = \sum_{k=0}^{\infty} \frac{\phi^{(k)}(x_0)}{k!}(x - x_0)^k. \tag{1.3.3}$$

The traditional higher-order Taylor series method needs high symbolic computations. But in DTM does not compute the derivatives symbolically. However, the these derivatives are computed recursively, which are defined by transformed equations of the original functions. In practice, the approximate value of a function $\phi(x)$ can be expressed as

$$\phi(x) = \sum_{k=0}^{\ell} \Phi(k)(x - x_0)^k, \tag{1.3.4}$$

and so, DTM is based on Taylor series expansion, in which the approximate solutions are obtained in a polynomial form [301]. The readers are referred to [3,155,181,184,205,288,301] and the reference therein for further details on DTM.

The DTM is generalized first by Chen [42] for partial differential equations. The generalized differential transform of a functions $\theta(x, y)$ is defined by

$$\Theta(h,k) = \frac{1}{h!k!}\left(\frac{\partial^{h+k}\theta(x,y)}{\partial x^k \partial y^k}\right)_{x=x_0,y=y_0} \tag{1.3.5}$$

and the differential inverse transform of $F(k)$'s is defined by

$$\theta(x,y) = \sum_{h=0}^{\infty}\sum_{k=0}^{\infty}\Theta(h,k)(x-x_0)^h(y-y_0)^k \tag{1.3.6}$$

From (1.3.5) and (1.3.6), we get

$$\theta(x,y) = \sum_{h=0}^{\infty}\sum_{k=0}^{\infty}\frac{(x-x_0)^h}{h!}\frac{(y-y_0)^k}{k!}\left(\frac{\partial^{h+k}\theta(x,y)}{\partial x^k \partial y^k}\right)_{x=x_0,y=y_0}. \tag{1.3.7}$$

In [185], Odibat and Momani developed fractional DTM, which is based on 2 dimensional DTM [19,269], generalized Taylor's formula [183] and Caputo fractional derivative [39], the proposed 2D DTM was further extended in the study of fractional differential equations.

**B. Fractional reduced differential transform method**

Using basic properties 1-dim DTM, a function $\psi$ of two variables with property $\psi(x,t) = f(x)g(t)$ can be embodied as

$$\psi(x,t) = \sum_{\ell=0}^{\infty} F_\alpha(\ell)\, t^{\alpha\ell}\sum_{j=0}^{\infty} G(j)x^j = \sum_{\ell=0}^{\infty}\Psi_\alpha^\ell(x)t^{\alpha\ell}, \tag{1.3.8}$$

where $0 < \alpha \leq 1$, $\Psi_\alpha^\ell(x) := \sum_{j=0}^{\infty} G(j)F_\alpha(\ell)x^j$ is the spectrum of $\psi(x,t)$. Throughout the thesis $\Psi_\alpha^\ell(x)$ or $\Psi_\ell(x)$ (uppercase) is used for the fractional reduced transformed function of $\psi(x,t)$ (lowercase). Some basic definitions and properties of FRDTM [15,252] are described in the following

**Definition 1.3.1.** *Let $\psi(x,t)$ be an analytic and continuously differentiable, then*

(a) *FRDT or the spectrum of $\psi$ is given by*

$\Psi_\alpha^k(x) = \frac{1}{\Gamma(k\alpha+1)} \left[\mathcal{D}_t^{\alpha k} \left(\psi\left(x,t\right)\right)\right]_{t=t_0}, \quad k = 0, 1, \ldots$

(*b*) *The inverse FRDT of* $\Psi_\alpha^k(x)$ *is defined by*

$$\psi\left(x,t\right) = \lim_{m\to\infty} \overline{s}_m = \sum_{k=0}^{\infty} \Psi_\alpha^k(x) \left(t-t_0\right)^{k\alpha}.$$

*where* $\overline{s}_m = \sum_{k=0}^{m} \Psi_\alpha^k(x) \left(t-t_0\right)^{k\alpha}$ *denotes the approximate solution with first* $m$ *iterations.*

*In particular for* $t_0 = 0$, *we get* $\psi\left(x,t\right) = \sum_{k=0}^{\infty} \Psi_\alpha^k(x) t^{k\alpha}$.

Let $X = (x_1, x_2, \ldots, x_n)$ be a vector of $n$ variables. Some basis properties of FRDTM are listed below

**Theorem 1.3.2.** *[15, 247] Let* $\Psi_\alpha^k\left(X\right)$ *and* $\Phi_\alpha^k\left(X\right)$ *be the spectrums of the analytic and continuously differentiable functions* $\psi(X,t)$ *and* $\phi(X,t)$, *respectively, and*

(1) *If* $\theta(X,t) = \ell_1\psi\left(X,t\right) \pm \ell_2\phi\left(X,t\right)$, *then*

$$\Theta_\alpha^k(X) = \ell_1\Psi_\alpha^k\left(X\right) \pm \ell_2\Phi_\alpha^k\left(X\right).$$

(2) *If* $\theta(X,t) = \psi\left(X,t\right)\phi\left(X,t\right)$, *then* $\Theta_\alpha^k(X) = \sum_{r=0}^{k} \Psi_\alpha^r(X)\Psi_\alpha^{k-r}(X)$.

(3) *If* $\theta(x,t) = f(X)\psi(X,t)$, *then*

$$\Theta_\alpha^k(X) = f(X)\Psi_\alpha^k\left(X\right).$$

(4) *If* $\theta(X,t) = x_i^m t^n \psi\left(X,t\right)$, *then* $\Theta_\alpha^k(X) = \begin{cases} x_i^m \Psi_\alpha^{k\alpha-n}\left(X\right) & \text{if } k\alpha \geq n \\ 0, & \text{else.} \end{cases}$

(5) *If* $\theta(X,t) = \mathcal{D}_{x_i}^{\beta}\psi\left(X,t\right)$; $\phi(X,t) = x_i^m t^\gamma$. *Then*

$$\Theta_\alpha^k(X) = \mathcal{D}_{x_i}^{\beta}\Psi_\alpha^k\left(X\right); \qquad \Phi_\alpha^k(X) = x_i^m\delta(k\alpha-\gamma); \qquad \delta(k) = \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{otherwise} \end{cases}$$

(6) *If* $\theta(X,t) = D_{x_i}^{r_i} D_t^{n\alpha} \psi(X,t)$. *Then* $\Theta_\alpha^k(X) = \frac{\Gamma(1+(k+n)\alpha)}{\Gamma(1+k\alpha)} D_{x_i}^{r_i} \Psi_\alpha^k(X)$. *In particular,*

(a) *If* $\theta(X,t) = D_{x_i}^{r_i} \psi(X,t)$. *Then* $\Theta_\alpha^k(X) = D_{x_i}^{r_i} \Psi_\alpha^k(X)$.

(b) *If* $\theta(X,t) = D_t^{n\alpha} \psi(X,t)$. *Then* $\Theta_\alpha^k(X) = \frac{\Gamma(1+(k+n)\alpha)}{\Gamma(1+k\alpha)} \Psi_\alpha^k(X)$.

(7) *If* $\theta(X) = \left.\frac{\partial^\beta \psi(X,t)}{\partial t^\beta}\right|_{t=0}$, *then*

$$\Theta_\alpha^r(X) = \begin{cases} \Psi_\alpha^r(X) & \text{if } r\alpha = m \\ 0 & \text{otherwise} \end{cases}, r = 0, 1, \ldots \frac{\beta}{\alpha} - 1.$$

Fractional reduced differential transform with $\alpha = 1$ is said to be reduced differential transform [267, 292], which was proposed first by Keskin [110]. After the seminal work of Keskin [110, 111], the reduced differential transform method has been widely employed to solve different type of classical and fractional partial differential equations [79, 162, 204, 211, 251, 255, 256] and many more.

## 1.4 Numerical Techniques

Most of the cases, it is not easy to predict the precise solution behavior of every PDEs model analytically, and so, the development of numerical techniques to investigate the predictions of PDEs models of such phenomena becomes very popular among the researchers. In general, the approximate numerical solutions are obtained in terms of the functional values at certain discrete points (grid/mesh points). In most of the numerical techniques, a bridge seems between the derivatives & the functional values at certain meshes. Many numerical techniques have been developed in the literature, among them, finite difference technique, finite element technique, and finite volume techniques are considered in the category of low order techniques while the spectral/pseudo-spectral technique is taken as global technique. The method of weighted residuals is the generalization of spectral and pseudo-spectral techniques.

A brief description of compact finite difference schemes and differential quadrature methods have been reported in the following

## 1.5 Finite difference method

One of the simplest methods, the finite difference methods (FDMs) were introduced by Euler in 1768 for one-dimensional differential equations and were probably extended in two dimensions by Runge in 1908. FDMs are very popular, which are computed from Taylor series or a polynomial approximation. FDMs as tools for solving partial differential equations are ease of implementation and flexibility concerning the boundary conditions. In FDM, the partial derivatives are replaced by finite difference approximations, and so, the PDEs is reduced to a large algebraic system of equations. Thus, the PDEs are converted into discrete quantities of dependent/independent variables, resulting in simultaneous algebraic equations with all unknowns prescribed at discrete mesh/grid points for the entire domain. Appropriate methods of solution are chosen in different applications, e.g., in fluid dynamics applications, depending upon the particular physics of the flows (which may include inviscid, viscous, incompressible, compressible, irrotational, rotational, laminar, turbulent, supersonic or hypersonic flows) FDMs are proposed to conform to these different physical phenomena.

The formulation of FDMs for one-dimensional problem is easy but for multidimensional problems, meshes must be structured in either two or three dimensions and curved meshes must be transformed into orthogonal Cartesian meshes. The challenge in analyzing finite difference methods for new classes of problems are often to obtain an appropriate definition of stability that allows one to show the convergence and to compute the errors in the approximation. FDMs discretize the governing PDEs directly using their strong form. It is the most straightforward way to obtain the discrete system equations but large grid distortions need to be avoided, the typical boundary conditions can't be handled easily. These schemes cannot easily be applied to very complex flow geometry shapes.

#### Compact finite difference scheme

The large stencils are cumbersome near the edge of the domain, and so, a significant disadvantage of the finite difference method is the widening of the computational stencil as the order of the approximation is increased. Luckily, the high-order finite difference meth-

ods can be derived with compact stencils (and so-called compact finite difference methods/schemes) at the expense of a small complication in their evaluation. The high order compact schemes are implicit in nature and provide high order accuracy and better resolution characteristics for the same number of grid points in comparison to the classical finite difference methods with the same number of grid points [152]. This feature brings them closer to the spectral methods while the freedom in choosing the mesh geometry and the boundary conditions (as in finite difference schemes) is maintained.

In the past years, various approaches of compact finite difference schemes for the approximations of the derivatives are derived on uniform grid [7, 46, 95] and nonuniform grids [47, 66, 72]. Chu and Fan developed sixth/eighth order three-point compact finite difference methods on uniform grids in [46] and extended to non-uniform grids in [47] and many more. Lele derived various order compact finite difference schemes for first and second derivatives using Fourier analysis approach [152], are generalizations of the Padé schemes. Moreover, he extended compact finite difference schemes for third/fourth order derivatives [152]. High order compact finite difference schemes have been extensively studied and widely applied to study the problems involving incompressible, compressible, and hypersonic flows [278]), computational aeroacoustic [43]) and many more. We refer the readers to [152, 165, 231] for further details on compact finite difference schemes for various order derivatives.

In the following compact finite difference schemes for first and second order derivatives as derived in [152] are reported for uniform grids $x_i = h(i-1)$ for $1 \leq i \leq N$, read $\phi_i = f(x_i)$ as the functional value of $f$ at grid $x_i$, $i \in \Delta_N$.

### A. Approximation for the first derivative

The finite difference approximation for the first derivative $\phi'_i = \frac{d\phi}{dx}$ at $x_i$ depends on the function values near to $x_i$ grid, for instance, second and fourth order central differences the approximation of $\phi'_i$ depends on the sets $(\phi_{i-1}, \phi_{i+1})$ and $(\phi_{i-2}, \phi_{i-1}, \phi_{i+1}, \phi_{i+2})$ of functional values, respectively. Lele [152] derived compact finite difference schemes for first order

derivative from the following approximation

$$\beta\phi'_{i-2}+\alpha\phi'_{i-1}+\phi'_i+\alpha\phi'_{i+1}+\beta\phi'_{i+2}=C\frac{\phi_{i+3}-\phi_{i-3}}{6h}+B\frac{\phi_{i+2}-\phi_{i-2}}{4h}+A\frac{\phi_{i+1}-\phi_{i-1}}{2h} \quad (1.5.1)$$

The values of $A, B, C$ and $\alpha, \beta$ or the relations between these coefficients are derived by matching the coefficients of various orders in Taylor series expansion, where the first unmatched coefficient defines the formal truncation error of the approximation (1.5.1), see [152, Eq.(2.1.1)-Eq.(2.1.5)].

($a$) second order: $A + B + C = 1 + 2\alpha + 2\beta$.

($b$) fourth order: $A + 2^2B + 3^2C = 2\frac{3!}{2!}(\alpha + 2^2\beta)$, and so on.

The general relation (1.5.1) with certain conditions from [152, Eq.(2.1.1)-Eq.(2.1.5)] leads to a variety of compact finite difference schemes for first derivative. A variety of tridiagonal compact finite difference schemes are derived by restricting $\beta = 0$ while for $\beta \neq 0$ generates pentadiagonal schemes, e.g., relation (1.5.1) with ($a$) and ($b$) leads to a three-parameter family of fourth-order schemes. Moreover, a two-parameter family of sixth-order pentadiagonal schemes is obtained by imposing the additional constraint of sixth order formal accuracy. Among them, some tridiagonal compact finite difference schemes are given below

**Fourth order accurate compact approximations:** The truncation error $T.E.$ and coefficients for one parameter $\alpha$ family of fourth order tridiagonal schemes is

$$T.E. = \frac{4}{5!}(3\alpha - 1)h^4\phi^{(5)} \;\Big|\; A = \frac{2}{3}(\alpha + 2), B = \frac{1}{3}(4\alpha - 1), C = 0, \beta = 0. \quad (1.5.2)$$

with maximum stencil size $\ell = 3$ on left side and $m = 5$ on the right. Scheme (1.5.2) with $\alpha = 0$ reduces to the well-known fourth order central difference scheme. Moreover, scheme (1.5.2) with $\alpha = \frac{1}{4}$ is the classical Padé scheme, and so, it is the generalization of Padé scheme. Scheme (1.5.2) with $\alpha = \frac{1}{3}$ reduces to sixth order accurate scheme of the form

$$T.E. = \frac{4}{7!}h^6\phi^{(7)} \;\Big|\; \alpha = \frac{1}{3},\ \beta = 0,\ A = \frac{14}{9},\ B = \frac{1}{9}, C = 0. \quad (1.5.3)$$

The specific tridiagonal compact finite difference schemes correspond to $\alpha = \frac{1}{4}$ and $\alpha = \frac{1}{3}$ were proposed in [53, pp. 538]. A family of sixth order compact finite difference scheme is defined in the following

**Sixth order accurate compact approximations:** The truncation error $T.E.$ and coefficients for one parameter $\alpha$ family of sixth order tridiagonal compact finite difference schemes are given by

$$T.E. = \frac{12}{7!}(3-8\alpha)h^6\phi^{(7)} \;\Big|\; \beta = 0, \quad A = \frac{1}{6}(\alpha+9), \quad B = \frac{1}{15}(32\alpha-9), \quad C = \frac{1}{10}(-3\alpha+1). \tag{1.5.4}$$

with maximum stencil sizes $\ell = 3;\ m = 7$. Sixth order tridiagonal scheme (1.5.3) is the special case of the family (1.5.4) with $\alpha = \frac{1}{3}$. In particular for $\alpha = \frac{3}{8}$, the family (1.5.4) of sixth order schemes can be further specialized into an eighth order tridiagonal scheme ($\beta = 0$) derived from (1.5.1) of the highest formal accuracy. It is noticed that Padé approximations of the first derivative are the special cases of the family of compact finite difference schemes derived from (1.5.1), see [122, 152]. For further details on the higher-order tridiagonal/pentadiagonal compact finite difference schemes with accuracy up to tenth-order, see [152, Table 1]

### B. Approximation for the second derivative

Analogous to the first derivative, Lele [152] derived compact finite difference schemes for second-order derivative from the following approximation

$$\begin{aligned}\beta\phi''_{i-2} + \alpha\phi''_{i-1} + \phi''_i + \alpha\phi''_{i+1} + \beta\phi''_{i+2} \\ = C\frac{\phi_{i+3} - 2\phi_i + \phi_{i-3}}{9h^2} + B\frac{\phi_{i+2} - 2\phi_i + 2\phi_{i-2}}{4h^2} + A\frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{h^2},\end{aligned} \tag{1.5.5}$$

where $\phi''_i$ denotes finite difference approximation of the second derivative of $\phi$ at node $x_i$. The values of $A, B, C$ and $\alpha, \beta$ or the relations between these coefficients are derived by matching the coefficients of various orders in Taylor series expansion, where the first unmatched coefficient defines the formal truncation error of the approximation (1.5.5),

see [152, (2.2.1)-(2.2.5)].

A variety of compact finite difference schemes with their maximum stencils sizes and truncation errors for second derivative within (1.5.5) as reported in [152, Table 2] are given in the following

**Fourth order tridiagonal compact finite difference schemes:** The truncation error $T.E.$ and coefficients for $\alpha$ family of fourth order tridiagonal compact finite difference schemes are as follows:

$$T.E. = -\frac{4}{6!}(11\alpha - 2)h^4\phi^{(6)} \;\Big|\; A = \frac{4}{3}(1-\alpha),\ B = \frac{-1}{3}(-1+10\alpha),\ C = 0,\ \beta = 0. \quad (1.5.6)$$

The specified scheme (1.5.6) with $\alpha \to 0$ is the well known fourth order central difference scheme while scheme (1.5.6) with $\alpha = \frac{1}{10}$ is the well known classical Padé scheme.

**Sixth order tridiagonal compact finite difference schemes:** The specified case of scheme (1.5.6) for $\alpha = \frac{2}{11}$ become a sixth order tridiagonal scheme are as follows:

$$T.E. = -\frac{8}{11!}h^6\phi^{(8)} \;\Big|\; \alpha = \frac{2}{11},\ \beta = 0, \quad A = \frac{12}{11}, \quad B = \frac{3}{11}, \quad C = 0. \quad (1.5.7)$$

The specified cases of scheme (1.5.6) for $\alpha = \frac{1}{10}$ and $\alpha = \frac{2}{11}$ were proposed in [53, pp. 538]

**Fourth order pentadiagonal compact finite difference schemes:** The truncation error $T.E.$ and coefficients for three-parameter family of fourth-order compact finite difference schemes derived from (1.5.5) with $\beta \neq 0$, $c \neq 0$ are as follows:

$$\begin{aligned} T.E. &= -\frac{4}{6!}(-2 + 11\alpha - 124\beta + 20\ C)h^4\phi^{(8)}; \\ A &= \frac{4 - 4\alpha - 40\beta + 5C}{3},\ b = \frac{-1 + 10\alpha + 46\beta - 8C}{3} \end{aligned} \quad (1.5.8)$$

The above class of compact finite difference schemes is the generalization of the following two parameter family of sixth order compact finite difference schemes, one parameter family of eighth order compact finite difference schemes and a tenth order compact finite difference scheme.

**Sixth order pentadiagonal compact finite difference schemes:** The truncation

error $T.E.$ and coefficients for two-parameter family of sixth order compact finite difference schemes, obtained by imposing sixth order constraints on (1.5.8), are as follows:

$$\begin{aligned} T.E. &= \frac{-8}{8!}(9-38\alpha+214\beta)h^6\phi^{(8)}; \\ A &= \frac{6-9\alpha-12\beta}{4},\ B = \frac{-3+24\alpha-6\beta}{5},\ C = \frac{2-11\alpha+124\beta}{20} \end{aligned} \tag{1.5.9}$$

**Eighth order pentadiagonal compact finite difference schemes:** The truncation error $T.E.$ and coefficients for one parameter family of eighth order compact finite difference schemes, obtained by imposing eighth order constraints on (1.5.9), are as follows:

$$\begin{aligned} T.E. &= \frac{899\alpha-334}{2696400}h^8\phi^{(10)}; \\ \beta &= \frac{38\alpha-9}{214},\ A = \frac{696-1191\alpha}{428},\ B = \frac{2454\alpha-294}{535},\ C = \frac{1179\alpha-344}{2140} \end{aligned} \tag{1.5.10}$$

The specified case of the eighth-order scheme (1.5.10) with $\alpha = \frac{344}{1179}$, i.e., $c = 0$ was proposed in [53, pp.539].

**Tenth order pentadiagonal compact finite difference scheme:** The truncation error $T.E.$ and coefficients for a tenth order compact finite difference scheme, obtained by imposing tenth order constraints on (1.5.10), are as follows:

$$T.E. = \frac{619}{299043360}h^{10}\phi^{(12)}\Bigg|\ \beta = \frac{43}{1798},\ \alpha = \frac{334}{899},\ a = \frac{1065}{1798},\ b = \frac{1038}{899},\ \frac{79}{1798} \tag{1.5.11}$$

which is the highest formal accuracy within the class of schemes (1.5.5).

## 1.6 Differential Quadrature Method: a survey

This section deals with the introductory information on the differential quadrature method (DQM) developed by Bellman et al. (1972). Before describing differential quadrature, first, we revisit the integral quadrature in the following

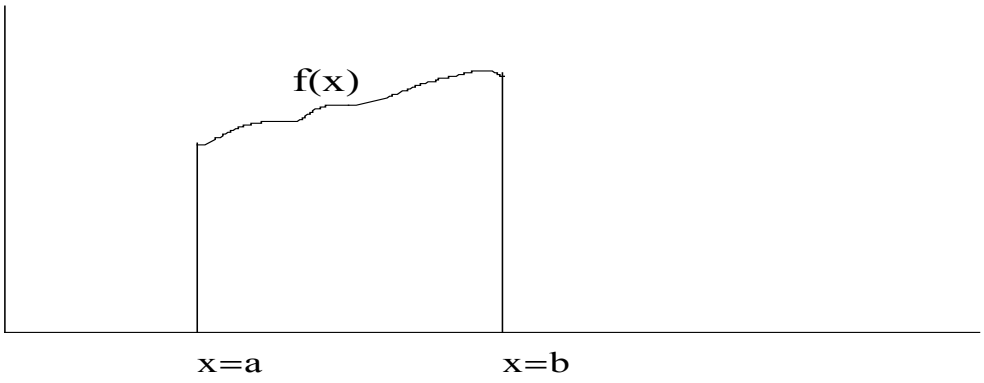


Figure 1.1: Integrating $f(x)$ over an interval

### 1.6.1 Integral Quadrature

Most of the case, the most critical problem in the fields of science and engineering is to evaluate $I = \int_a^b f(x)dx$ over a finite interval $[a, b]$. The value of the integral $I$ is $F(b) - F(a)$ whenever $\exists$ a function $F$ with the property that $dF = fdx$. Practically, it very difficult to identify an explicit expression for $F$ for each problem. Indeed, it is impossible to identify the function $F$ whenever the values of $f$ may known at a set of certain grids only. This shows that there is an essential need of a numerical technique to compute the value of the integral in such cases. Noticed that the integral $I = \int_a^b f(x)dx$ is the approximation of the area of the region bounded by $x$ axis and $f(x)$ between $x = a$ and $x = b$ as shown in Figure 1.1. Based on this principle, many vigorous numerical techniques have been developed in the past years. Let $P[a, b] = \{x_i \in [a, b] : a = x_1 < x_2 < \ldots < x_{N-1} < x_N = b\}$ be any partition of $[a, b]$. Setting $\Delta_N = \{1, 2, \ldots, N\}$ and the functional value of $f$ at each grid point $x_\ell$ by $f_\ell = f(x_\ell)$, then the integral quadrature formula for $\int_a^b f(x)dx$ is given as

$$\int_a^b f(x)dx = \sum_{\ell \in \Delta_N} \omega_\ell f_\ell,$$

where each $\omega_\ell, \ell \in \Delta_n$ is the weighting coefficient. Thus, in integral quadrature formula an integral $\int_a^b f(x)dx$ is approximated as the weighted linear sum of functional values at each grid over the integral domain.

If the grid points in the partition $P[a, b]$ is distributed uniformly, that is, $h_x = x_{i+1} - x_i = \frac{b-a}{N-1},\ i \in \Delta_N \setminus \{N\}$. Then many well known integral quadrature formula like *Trapezoidal*

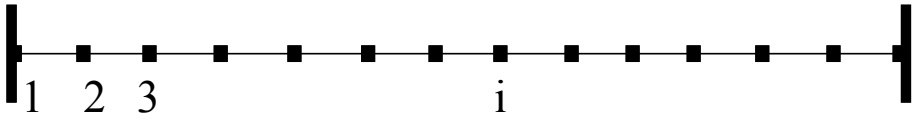


Figure 1.2: Partition of the computational domain $\Omega_1$ with N grid points

*formula, Simpson's formula and many more* are available in the literature.

### 1.6.2 Differential quadrature method

Analogous to integral quadrature, the differential quadrature was introduced first by Bellman et al. (1972). The differential quadrature method (DQM) is a numerical technique for the approximation of the derivatives of a smooth function at the certain grid points over the computational domain. The description of DQM for one-dimensional problem is as follows

Let $\phi(x)$ be sufficiently smooth function over the computational domain $\Omega_1 = [a, b]$. Then the approximate value of the $r$th order derivative

$$\phi_i^{(r)} = \left.\frac{d^r\phi}{dx^r}\right|_{x=x_i}$$

of $\phi(x)$ with respect to $x$ at grid $x_i$ is approximated as the weighted linear sum of all the functional values of $\phi(x)$ in the whole computational domain as follows

$$\phi_i^{(r)} = \sum_{\ell\in\Delta_N} a_{i\ell}^{(r)}\phi_\ell, \quad i \in \Delta_N \tag{1.6.1}$$

where each $a_{i\ell}^{(r)}$, $\ell, i \in \Delta_N$ is the weighting coefficient of $r$th order derivative and $N$ is total grid points in the partition of the computational domain $\Omega_1$, see Figure 1.6.2. $\phi_\ell \equiv \phi(x_\ell)$, the functional value at grid $x_\ell$. The expression (1.6.1) is referred to as the differential quadrature. The computation of the weighting coefficients $a_{i\ell}^{(r)}$, $i, \ell \in \Delta_N$, in the differential quadrature is a key procedure. Once the weighting coefficients are determined, the bridge to the link the derivatives in the governing differential equation and the functional value at the each mesh point is established. Before developing DQM, the most of the numerical problems in the fields of science and engineering are studied via low order finite difference

method, finite element method and many more by considering a large number of grid points. The DQM is very standard method to compute very accurate numerical results for considerably smaller number of grid points, and so, the DQM requiring relatively little computational effort. The differential quadrature method (DQM) has been employed very accurately, efficiently and reliably on the variety of problems in the fields of engineering and sciences. After the seminal paper of Quan and Chang [196,197] and Bellman et al., the DQM has been implemented for various type of set of base functions, among others, cubic B-spline differential quadrature methods [124, 128, 129], Fourier expansion and Harmonic function based DQM [226, 228], Polynomial DQM [124, 131], quartic B-spline based DQM [24, 133], Quartic and quintic B-spline based DQM [130], exponential cubic B-spline based DQM [126], sinc DQM [134], generalized DQM [223], exponential cubic B-spline based DQM [126] and modified cubic B-spline based DQM [13,239–241]. The readers are suggested to see [29, 45, 50, 157, 158, 225] for a variety of problems studied numerically by DQM.

## 1.7 Computation procedure of weighting coefficient $a_{ij}^{(1)}, i, j \in \Delta_N$

Consider a partition $P[a, b]$ with $N$ grid points $P[a, b] = \{x_i \in [a, b] : a = x_1 < x_2 < \ldots < x_N = b\}$ of the computational domain $[a, b]$ of the problem. According to Bellman et al. [27], the approximation for the first order derivative $\phi_i^{(1)}$ of a sufficiently smooth function $\phi$ at $i$th grid point is direct from (1.6.1) with $r = 1$ as follows

$$\phi_i^{(1)} = \sum_{j \in \Delta_N} a_{ij}^{(1)} \phi_j, \quad i \in \Delta_N \tag{1.7.1}$$

Among the various approaches to compute the weighting coefficients $a_{ij}^{(1)},\ i, j \in \Delta_N$, some rigorous approaches are described in the following

### 1.7.1 Bellman's approach

Based on two different set of base functions, Bellman et al. [27] developed the following two approaches to compute the weighting coefficients $a_{ij}^{(1)},\ i, j \in \Delta_N$.

### 1.7.2 Bellman's first approach

This approach deals with the polynomial base functions to compute $a_{ij}^{(1)}, i, j \in \Delta_N$

$$s_k(x) = x^{k-1}, k \in \Delta_N. \tag{1.7.2}$$

On implementing the base functions (1.7.2) in equation (1.7.1) for $i$th grid point ($i \in \Delta_N$) yields the following system of equations

$$\begin{aligned} &\sum_{j\in\Delta_N} a_{ij}^{(1)} = 0, \\ &\sum_{j\in\Delta_N} a_{ij}^{(1)} x_j = 1 \\ &\sum_{j\in\Delta_N} a_{ij}^{(1)}\ x_j^{k-1} = (k-1)x_i^{k-2}, \quad k \in \Delta_N \setminus \{1, 2\}. \end{aligned} \tag{1.7.3}$$

Being the coefficient matrix of the system (1.7.3) is of Vandermonde form and its order is $N \times N$, the above system has unique solution for $a_{ij}^{(1)},\ i, j \in \Delta_N$. Regrettably, the matrix is ill-conditioned for large $N$, and so, its inversion is too difficult. In general $N$ is chosen less than 13.

### 1.7.3 Bellman's second approach

This approach is similar to the first approach but main difference is that the following set of test functions are used with DQM as base functions

$$s_\ell(x) = \frac{L_N(x)}{(x - x_\ell)L_N^{(1)}(x_\ell)}, \ell \in \Delta_N, \tag{1.7.4}$$

where $L_N^{(1)}(x)$ is the first order derivative of Legendre polynomial $L_N(x)$ of degree $N$. By choosing the roots $x_\ell, \ell \in \Delta_N$ of the shifted Legendre polynomial as the grid points, Bellman et al. [27] computed the following simple algebraic formulae to compute $a_{ij}^{(1)},\ i, j \in \Delta_N$

$$\begin{cases} a_{ij}^{(1)} = \dfrac{L_N^{(1)}(x_i)}{(x_i - x_j)(L_N^{(1)})(x_j)}, & \text{i} \neq j,\ i, j \in \Delta_N, \\ a_{ii}^{(1)} = \dfrac{(1 - 2x_i)}{2x_i(x_i - 1)}, & \text{i} = j,\ i, j \in \Delta_N. \end{cases} \tag{1.7.5}$$

It is worth mentioning that the computation of the weighting coefficients from equation (1.7.5) is not an easy task. Since the coordinates of the grid points cannot be chosen arbitrarily. Instead, one should choose the roots of the shifted Legendre polynomial of degree $N$ as the grid points in the partition of the computational domain of the problem. This is why the second approach is not as flexible as the first one. Due to the inflexibility associated with this approach in selecting the grid points, the first approach is usually adopted for the numerical study of the variety of physical problems.

### 1.7.4 Quan and Chang's approach

After Bellman's approaches for computing the weighting coefficients, many attempts have been made in various articles by researchers to improve the Bellman's approaches in computing the weighting coefficients. One of the most useful approaches was introduced by Quan and Chang [196, 197] by adopting the following set of base functions with Lagrange interpolation polynomials

$$s_k(x) = \frac{M(x)}{(x - x_k)M^{(1)}(x_k)}, \quad k \in \Delta_N, \tag{1.7.6}$$

where

$$\begin{cases} M(x) = \displaystyle\prod_{\ell \in \Delta_N} (x - x_\ell) \\ M^{(1)}(x_i) = \displaystyle\prod_{\ell \in \Delta_N \setminus \{i\}} (x_i - x_\ell). \end{cases} \tag{1.7.7}$$

Subsequently, on implementing the set of base functions (1.7.6) in any partition of the computational domain $[a, b]$ with $N$ grid points, Quan and Chang obtained the following algebraic formulations for computing the weighting coefficients $a_{ij}^{(1)},\ i, j \in \Delta_N$

$$\begin{cases} a_{ij}^{(1)} = \dfrac{1}{(x_j - x_i)} \displaystyle\prod_{k \in \Delta_N \setminus \{i,j\}} \dfrac{(x_i - x_k)}{(x_j - x_k)}, & \mathrm{i} \neq j,\ i, j \in \Delta_N, \\ a_{ii}^{(1)} = \displaystyle\sum_{k \in \Delta_N \setminus \{i\}} \dfrac{1}{x_i - x_k}, & \mathrm{i} = j,\ i, j \in \Delta_N. \end{cases} \tag{1.7.8}$$

It is worth mentioning that contrary to Bellman's second approach, this approach has no restriction for the selection of the grid points.

### 1.7.5 Shu's general approach [225]

Inspired by Bellman's approaches, Shu has developed a general approach which covers all the approaches including Quan and Chang's approach. In his approach, Shu considered two typical sets of the base polynomials as listed in (1.7.9a)-(1.7.9b). Shu demonstrate that computation of weighting coefficients by all the above approaches is same due to polynomial approximation and linear vector space analysis.

$$s_\ell(x) = \frac{M(x)}{(x - x_\ell) M^{(1)}(x_\ell)}, \quad \ell \in \Delta_N \tag{1.7.9a}$$

$$s_\ell(x) = x^{\ell - 1}, \qquad \ell \in \Delta_N \tag{1.7.9b}$$

where $M(x)$ and $M^{(1)}(x_i)$ are same as defined in equation (1.7.7) but the computation procedure of the weighting coefficients is different. For simplicity, we set

$$\begin{cases} M(x) = N(x, x_\ell)(x - x_\ell), & \ell \in \Delta_N \\ N(x_i, x_j) = \delta_{ij}\ M^{(1)}(x_i), & i, j \in \Delta_N, \end{cases} \tag{1.7.10}$$

where $\delta_{ij}$ is well known Kronecker operator. Equation (1.7.10) and Equation (1.7.9a) yields

$$s_\ell(x) = \frac{N(x, x_\ell)}{M^{(1)}(x_\ell)}, \quad \ell \in \Delta_N \tag{1.7.11}$$

Finally, equation (1.7.1) with test function (1.7.11) yields

$$a_{ij}^{(1)} = \frac{N^{(1)}(x_i, x_j)}{M^{(1)}(x_j)}, \quad i, j\Delta_N, \tag{1.7.12}$$

where $M^{(1)}(x_j)$ can be computed from equation (1.7.7). To compute $N^{(1)}(x_i, x_j)$, first differentiate successively relation (1.7.10) $r$ times with respect to $x$ to obtain the following

$$M^{(r)}(x) = (x - x_j)\ N^{(r)}(x, x_j) + r\ N^{(r-1)}(x, x_j), \quad j \in \Delta_N; r \geq 1 \tag{1.7.13}$$

where $M^{(r)}(x)$ and $N^{(r)}(x, x_j)$ denote $r$th order derivatives of $M(x)$ and $N(x, x_j)$, respectively. Equation (1.7.13) yields the following expression for $N^{(1)}(x_i, x_j)$

$$\begin{cases} N^{(1)}(x_i, x_j) = \dfrac{M^{(1)}(x_i)}{x_i - x_j}, & i \neq j,\ i, j \in \Delta_N \\ N^{(1)}(x_i, x_i) = \dfrac{M^{(2)}(x_i)}{2}, & i = j,\ i, j \in \Delta_N. \end{cases} \tag{1.7.14}$$

Equation (1.7.14) and (1.7.12) yields the following

$$a_{ij}^{(1)} = \frac{M^{(1)}(x_i)}{(x_i - x_j)M^{(1)}(x_j)}, \quad i \neq j,\ i, j \in \Delta_N \tag{1.7.15a}$$

$$a_{ii}^{(1)} = \frac{M^{(2)}(x_i)}{2M^{(1)}(x_i)}, \qquad i = j,\ i, j \in \Delta_N. \tag{1.7.15b}$$

It is worth mentioning that for given $x_i$, $M^{(1)}(x_i)$ can be computed easily from equation (1.7.7), and so, $a_{ij}^{(1)}$, $i \neq j, i, j \in \Delta_N$ can be computed easily from equation (1.7.15). However, the computation of $a_{ii}^{(1)}$, $i \in \Delta_N$ is based on the computation of second-order derivative $M^{(2)}(x_i)$ is not a easy task. This difficulty was eliminated by adopting second set of base polynomials (1.7.9b). According to the property of a linear vector space, "if one set of base polynomials satisfies a linear operator, so does another set of base polynomials. As a consequence, the values of $a_{ij}^{(1)}, i, j \in \Delta_N$ obtained from the linear system derived from Lagrange interpolation polynomials equation (1.7.9a) are equivalent to that derived from another set of base polynomial $x^{k-1}, k \in \Delta_N$ in (1.7.9b). Hence, for base polynomial

$\phi(x) = 1$ for $k = 1$ in base functions (1.7.9b) with (1.7.1) yields the following

$$a_{ii}^{(1)} = -\sum_{j \in \Delta_N \setminus \{i\}} a_{ij}^{(1)}, \qquad i \in \Delta_N. \tag{1.7.16}$$

Thus, the weighting coefficient $a_{ij}^{(1)},\ i, j \in \Delta_N$ can be computed easily from equation (1.7.15a) and (1.7.16), which is derived from two sets of base polynomials in the linear polynomial vector space $V_N$.

## 1.8 Computation procedure of weighting coefficient $a_{ij}^{(2)}, i, j \in \Delta_N$ [225]

According to Bellman et al. [27], the approximation for second order derivative $\phi_i^{(2)}$ of a sufficiently smooth function $\phi$ at $i$th grid point is direct from (1.6.1) with $r = 2$ as follows

$$\phi_i^{(2)} = \sum_{j \in \Delta_N} a_{ij}^{(2)} \phi_j, \quad i \in \Delta_N \tag{1.8.1}$$

Mainly the following two approaches are used to compute the weighting coefficients $a_{ij}^{(2)},\ i, j \in \Delta_N$

### 1.8.1 Quan and Chang's approach [196, 197]

This approach used the Lagrange interpolation polynomials as a set of base functions in DQM, to compute the weighting coefficients $a_{ij}^{(2)},\ i, j \in \Delta_N$ as follows

$$\begin{cases} a_{ij}^{(2)} = \dfrac{2}{(x_j - x_i)} \left( \displaystyle\prod_{\ell \in \Delta_N \setminus \{i,j\}} \dfrac{x_i - x_\ell}{x_j - x_\ell} \right) \left( \displaystyle\sum_{l \in \Delta_N \setminus \{i,j\}} \dfrac{1}{x_i - x_l} \right), & \text{i} \neq j,\ i, j \in \Delta_N \\ a_{ii}^{(2)} = 2 \displaystyle\sum_{\ell \in \Delta_{N-1} \setminus \{i\}} \left\{ \dfrac{1}{x_i - x_\ell} \left( \displaystyle\sum_{l=\ell+1, l \neq i}^{N} \dfrac{1}{(x_i - x_l)} \right) \right\}, & \text{i} \neq j,\ i, j \in \Delta_N. \end{cases} \tag{1.8.2}$$

### 1.8.2 Shu's general approach for computing $a_{ij}^{(2)}, i, j \in \Delta_N$

Analogous to the weighting coefficients for the first order derivative, Shu's approach for the weighting coefficients for second order derivative is also based on the polynomial approximation and linear vector space analyses. The two sets of base polynomials as given in (1.7.9a)-(1.7.9b) is used. On using Equation (1.7.11) into Equation (1.8.1), we get

$$a_{ij}^{(2)} = \frac{N^{(2)}(x_i, x_j)}{M^{(1)}(x_j)}, \qquad i, j \in \Delta_N, \tag{1.8.3}$$

and the following expression for $N^{(2)}(x_i, x_j)$ is direct from Equation (1.7.13)

$$\begin{cases} N^{(2)}(x_i, x_j) = \dfrac{M^{(2)}(x_i) - 2N^{(1)}(x_i, x_j)}{x_i - x_j}, & i \neq j,\ i, j \in \Delta_N \\ N^{(2)}(x_i, x_i) = \dfrac{M^{(3)}(x_i)}{3}, & i = j,\ i, j \in \Delta_N. \end{cases} \tag{1.8.4}$$

Equation (1.8.4) and Equation (1.8.3) yields the following

$$\begin{cases} a_{ij}^{(2)} = \dfrac{M^{(2)}(x_i) - 2N^{(1)}(x_i, x_j)}{(x_i - x_j)M^{(1)}(x_j)}, & i \neq j,\ i, j \in \Delta_N \\ a_{ii}^{(2)} = \dfrac{M^{(3)}(x_i)}{3M^{(1)}(x_i)}, & i = j,\ i, j \in \Delta_N. \end{cases} \tag{1.8.5}$$

Moreover for $i \neq j$, equation (1.7.15) and Equation (1.8.5) yields

$$a_{ij}^{(2)} = 2a_{ij}\left(a_{ii} - \frac{1}{x_i - x_j}\right), \qquad i \neq j,\ i, j \in \Delta_N \tag{1.8.6}$$

Equation (1.8.6) shows that for $i \neq j$, the weighting coefficients $a_{ij}^{(2)}, i, j \in \Delta_N$ can be computed easily. On the other hand, the computation of the weighting coefficient $a_{ii}^{(2)}, i \in \Delta_N$ is a tough task as it involves the third order derivative $M^{(3)}(x_i)$ whose computation is not easy. This difficulty was eliminated by adopting second set of base polynomials (1.7.9b) and a property of linear vector space as used to compute $a_{ii}^{(1)}, i \in \Delta_N$. That is, base polynomial $\phi(x) = 1$ for $k = 1$ in base functions (1.7.9b) with (1.8.1) yields the

following

$$a_{ii}^{(2)} = -\sum_{j \in \Delta_N \setminus \{i\}} a_{ij}^{(2)}, \quad i \in \Delta_N. \tag{1.8.7}$$

It is worth mentioning that Shu's general approach was the first approach to compute the weighting coefficients $a_{ij}^{(2)}, i, j \in \Delta_N$ from Equations (1.8.6)-(1.8.7).

## 1.9 Shu's recursive approach for computing $a_{ij}^{(r)}, i, j \in \Delta_N$, $r \geq 2$ [225]

Using differential quadrature formula, the approximation for $r$th order derivative $\phi_i^{(r)}$ and $(r-1)$th order derivative $\phi_i^{(r-1)}$ of a sufficiently smooth function $\phi$ at $i$th grid point can be read from (1.6.1) as follows

$$\begin{cases} \phi_i^{(r)} = \sum\limits_{j \in \Delta_N} a_{ij}^{(r)} \phi_j, & 2 \leq r \leq N-1, i \in \Delta_N, \\ \phi_i^{(r-1)} = \sum\limits_{j \in \Delta_N} a_{ij}^{(r-1)} \phi_j, & 2 \leq r \leq N-1, i \in \Delta_N. \end{cases} \tag{1.9.1}$$

This approach is also based upon two sets of base polynomials to derive an explicit formulation to compute the weighting coefficients $a_{ij}^{(r)},\ i, j \in \Delta_N$. Equations (1.9.1) with the first set of base polynomials from (1.7.11) yield the following

$$a_{ij}^{(r-1)} = \frac{N^{(r-1)}(x_i, x_j)}{M^{(1)}(x_j)}, \quad i, j \in \Delta_N \tag{1.9.2}$$

$$a_{ij}^{(r)} = \frac{N^{(r)}(x_i, x_j)}{M^{(1)}(x_j)}, \quad i, j \in \Delta_N \tag{1.9.3}$$

Equation (1.9.2) can be re-written as follows

$$N^{(r-1)}(x_i, x_j) = a_{ij}^{(r-1)} M^{(1)}(x_j), \quad i, j \in \Delta_N. \tag{1.9.4}$$

On the other hand, recurrence relation (1.7.13), we have

$$N^{(r-1)}(x_i, x_i) = \frac{M^{(r)}(x_i)}{r}, \qquad i = j,\ i, j \in \Delta_N, \tag{1.9.5}$$

$$N^{(r)}(x_i, x_j) = \frac{M^{(r)}(x_i) - rN^{(r-1)}(x_i, x_j)}{x_i - x_j},\ i \neq j,\ i, j \in \Delta_N, \tag{1.9.6}$$

$$N^{(r)}(x_i, x_i) = \frac{M^{(r+1)}(x_i)}{r+1}, \qquad i = j,\ i, j \in \Delta_N. \tag{1.9.7}$$

Equation (1.9.5) and Equation (1.9.6) leads to

$$N^{(r)}(x_i, x_j) = \frac{r\left\{N^{(r-1)}(x_i, x_i) - N^{(r-1)}(x_i, x_j)\right\}}{x_i - x_j}, i \neq j,\ i, j \in \Delta_N, \tag{1.9.8}$$

which can be further simplified by using (1.9.4) as follows

$$N^{(r)}(x_i, x_j) = \frac{r\left(a_{ii}^{(r-1)} M^{(1)}(x_i) - a_{ij}^{(r-1)} M^{(1)}(x_j)\right)}{x_i - x_j}, i \neq j,\ i, j \in \Delta_N, \tag{1.9.9}$$

The recursive values of the weighting coefficients $a_{ij}^{(r)}$ can be obtained from Equation (1.9.4) after using Equations (1.9.3) and (1.7.15a) as follows

$$a_{ij}^{(r)} = r\left(a_{ij}^{(1)} a_{ii}^{(r-1)} - \frac{a_{ij}^{(r-1)}}{x_i - x_j}\right), \quad i \neq j,\ i, j \in \Delta_N, 2 \leq r \leq N-1. \tag{1.9.10}$$

The formulation for $a_{ii}^{(r)}$ can be obtained by using equation (1.9.7) into equation (1.9.3) as follows

$$a_{ii}^{(r)} = \frac{M^{(r+1)}(x_i)}{(r+1)M^{(1)}(x_i)}, \quad i = j, i, j \in \Delta_N; 2 \leq r \leq N-1. \tag{1.9.11}$$

It is worth mentioning that equation (1.9.10) offer an easy way to compute the weighting coefficients $a_{ij}^{(r)}$ for $i, j \in \Delta_N$, $1 \neq j$ for $r$th order derivatives. On the other hand, the computation of the weighting coefficient $a_{ii}^{(r)}, i \in \Delta_N$ is a very tough task as it involves the third order derivative $M^{(r+1)}(x_i)$ whose computation complexity is a very tough task. Analogously, this difficulty is eliminated employing property of linear vector space and a second set of base polynomials (1.7.9b). That is, base polynomial $\phi(x) = 1$ for $k = 1$ in

(1.7.9b) with first equation of (1.9.1) yields the following

$$a_{ii}^{(r)} = - \sum_{j \in \Delta_N \setminus \{i\}} a_{ij}^{(r)}. \tag{1.9.12}$$

## 1.10 Computation procedure of weighting coefficients for multi-dimensional problems

In practice, most of the physical phenomena are modeled in terms of two/three space dimensional problems, and so, an extension of the differential quadrature approximation $N$-dimensional case becomes necessary. This section deals with the extension of differential quadrature approximation for two-dimensional problems with regular domain. In [224], it is demonstrated that one-dimensional polynomial based DQM can be extended directly to multi-dimensional cases whenever the discretization domain is regular (i.e., rectangle, circle, etc.).

For convenience, consider a rectangular domain for a description of the procedure. Let $\phi(x, y)$ be a smooth function of two variables defined on a regular domain as in Figure 1.3. The discretization step along each horizontal line, the $x$ interval, is same and analogously

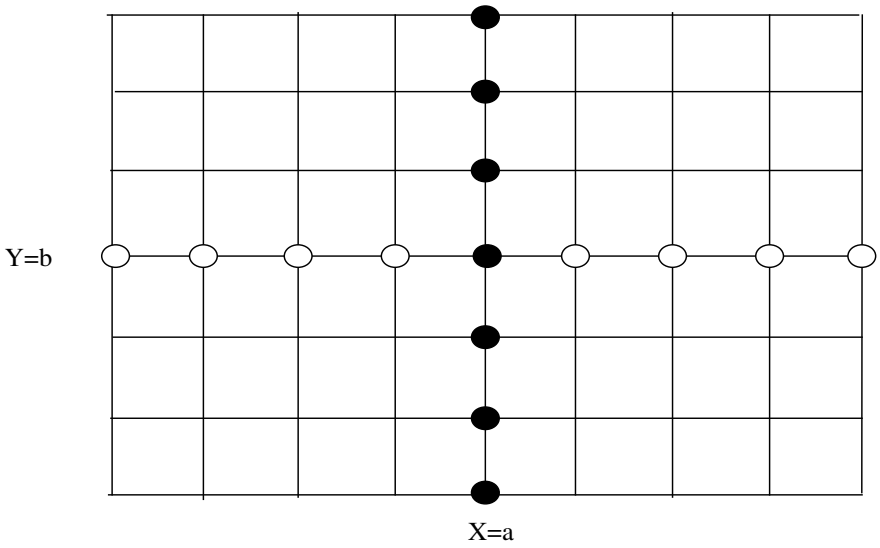


Figure 1.3: Uniform grid distribution in a regular (e.g., rectangular) domain

along each vertical line the $y$ interval, is the same. Therefore, one can assume the same $x$ coordinate distribution for each horizontal line while the same $y$ coordinate distribution for each vertical line. Let along $y = b$ for any constant $b$, the values of $\phi(x, b)$ at the grid points, denoted by the blank circles, can be approximated by a polynomial $P_N(x)$ of degree $(N-1)$ which constitutes an $N$-dimensional linear vector space $V_N$ with $N$ polynomials $r_i(x), i \in \Delta_N$ as basis vectors. Similarly, along the vertical line $x = a$ for any constant $a$, the values of $\phi(a, y)$ at the grid points, denoted by the dark circles, can be approximated by a polynomial $P_M(y)$ of degree $(M-1)$ which constitutes an $M$ dimensional linear vector space $V_M$ with $M$ polynomials $s_j(y), j \in \Delta_M$ as basis vectors. The value of $\phi$ at any grid point $(x, y)$ in the domain can be approximated as follows

$$\phi(x, y) \simeq P_{N\times M}(x, y) = \sum_{i\in\Delta_N} \sum_{j\in\Delta_M} \omega_{ij} x^{i-1} y^{j-1} \tag{1.10.1}$$

where $\omega_{ij}$ is a coefficient. It is to be noticed that $P_{N\times M}(x, y)$ forms an $N \times M$ dimensional linear polynomial vector space $V_{N\times M}$ with respect to the operation of vector addition and scalar multiplication, where $\tau_{ij}(x, y) = r_i(x)s_j(y), i \in \Delta_N, j \in \Delta_M$ is a basis vector of the linear vector space $V_{N\times M}$.

Let the values of $\phi$ at $(x_i, y_j)$ is read as $\phi_{ij} = \phi(x_i, y_j)$ for $i \in \Delta_N$, $j \in \Delta_M$. In DQM, the $r$-th order partial derivatives of $\phi(x, y)$ with respect to $x, y$ at the grid point $(x_i, y_j)$ can be approximated as follows

$$\begin{cases} \left.\dfrac{\partial^r \phi}{\partial x^r}\right|_{(x_i,y_j)} = \displaystyle\sum_{k\in\Delta_N} a_{ik}^{(r)} \phi_{kj} \quad i \in \Delta_N \\ \left.\dfrac{\partial^r \phi}{\partial y^r}\right|_{(x_i,y_j)} = \displaystyle\sum_{k\in\Delta_M} b_{jk}^{(r)} \phi_{ik} \quad j \in \Delta_M \end{cases} \tag{1.10.2}$$

where $a_{ij}^{(r)}$ and $b_{ij}^{(r)}$ are the weighting coefficients of $r$-th order partial derivatives of $\phi$ with respect to $x$ and $y$ respectively. By the properties of linear vector space, "if all the base polynomials $\tau_{ij}(x, y)$ satisfy the linear equation (1.10.2), so does any polynomial in $V_{N\times M}$"

Equation (1.10.2) with $r = 1$ for $\phi_{ij}(x, y)$ leads to

$$\begin{aligned} r_j^{(1)}(x_i) &= \sum_{\ell \in \Delta_N} a_{i\ell}^{(1)}\ r_j(x_\ell), \quad i, j \in \Delta_N \\ s_j^{(1)}(y_i) &= \sum_{\ell \in \Delta_M} b_{i\ell}^{(1)}\ s_j(y_\ell), \quad i, j \in \Delta_M \end{aligned} \tag{1.10.3}$$

where $r_j^{(1)}(x_i)$ be the first order derivative of $r_j(x)$ at $x_i$ and $s_j^{(1)}(y_i)$ be the first order derivative of $s_j(y)$ at $y_i$.

It is worth mentioning that the weighting coefficients $a_{ij}^{(1)}$ or $b_{ij}^{(1)}$, being related to only $r_j(x) or s_j(y)$ (see equation (1.10.3)), can be formulated directly from one dimensional case, i.e.,

$$\begin{cases} a_{ij}^{(1)} = \dfrac{M^{(1)}(x_i)}{(x_i - x_j)M^{(1)}(x_j)}, & i \neq j,\ i, j \in \Delta_N, \\ a_{ii}^{(1)} = -\displaystyle\sum_{j \in \Delta_N \backslash \{i\}} a_{ij}^{(1)}, & \mathrm{i} = j,\ i, j \in \Delta_N \\ b_{ij}^{(1)} = \dfrac{P^{(1)}(y_i)}{(y_i - y_j)P^{(1)}(y_j)}, & i \neq j,\ i, j \in \Delta_M, \\ b_{ii}^{(1)} = -\displaystyle\sum_{j \in \Delta_M \backslash \{i\}} b_{ij}^{(1)}, & i = j,\ i, j \in \Delta_M \end{cases} \tag{1.10.4}$$

where

$$M^{(1)}(x_i) = \prod_{j \in \Delta_N \backslash \{i\}} (x_i - x_j), \quad P^{(1)}(y_i) = \prod_{j \in \Delta_M \backslash \{i\}} (y_i - y_j)$$

The weighting coefficients $a_{ij}^{(r)}, i, j \in \Delta_N; b_{ij}^{(r)}, i, j \in \Delta_M$ for $r$-th order derivatives can be computed directly from Shu's recursive formula [225] based on polynomial DQM as follows

$$\begin{cases} a_{ij}^{(r)} = r\left(a_{ij}^{(1)} a_{ii}^{(r-1)} - \dfrac{a_{ij}^{(r-1)}}{x_i - x_j}\right), & \mathrm{i} \neq j, \mathrm{i}, j \in \Delta_N; \\ a_{ii}^{(r)} = -\displaystyle\sum_{j \in \Delta_N \backslash \{i\}} a_{ij}^{(r)}, & \mathrm{i} = j, \mathrm{i}, j \in \Delta_N; \\ b_{ij}^{(r)} = r\left(b_{ij}^{(1)} b_{ii}^{(r-1)} - \dfrac{b_{ij}^{(r-1)}}{y_i - y_j}\right), & \mathrm{i} \neq j, \mathrm{i}, j \in \Delta_M; \\ b_{ii}^{(r)} = -\displaystyle\sum_{j \in \Delta_M \backslash \{i\}} b_{ij}^{(r)}, & \mathrm{i} = j, \mathrm{i}, j \in \Delta_M; \end{cases} \tag{1.10.5}$$

## 1.11 Numerical study of time-dependent partial differential equation by employing DQM

The partial differential equations are of two types i) steady state (time-independent) and ii) unsteady state (time-dependent) differential equations. The DQM is applicable to solve both steady state equations and unsteady state equations. Most of the problems occurred in sciences and engineering are governed by unsteady state partial differential equations. This section deals with the study of unsteady state partial differential equation by employing DQM.

In general, DQM is used to approximate the spatial derivatives. In this way unsteady state partial differential equation reduces to unsteady state ordinary differential equations, which can be solved various existing approaches. Consider an unsteady state PDE of the form

$$\frac{\partial u}{\partial t} = L[u] + f(t), \tag{1.11.1}$$

associated with appropriate initial and boundary conditions. In Equation (1.11.1) $u(X,t)$ be unknown to be determine, $X = (x_1, x_2, \ldots, x_n)$ spatial coordinate in $n$-dimensional space, $f$ be a given function and $L$ is a differential operator containing all the spatial derivatives. DQM is used at each interior grid point to discretize the spatial derivatives in the differential operator $L[u]$. Thus, a set of first-order ODEs is obtained as follows

$$\frac{\partial U}{\partial t} = L[U] + F, \tag{1.11.2}$$

where $U$ is the vector denoting a set of unknown values at each interior point, $L[U]$ be a vector resulting from differential quadrature discretization, $F$ be a known vector arising from source term $f$ and the given boundary conditions. The resulting system of first order ODEs (1.11.2) with the initial value at each interior point, obtained from the given initial values, can be solved either via any rigorous time integration technique or by employing any well known explicitly or implicitly low order finite difference method.

## 1.12 Literature review

The classical Navier-stokes Equation can be regarded as Newton's second law of motion for fluid substances which is a combination of momentum equation, continuity equation and energy equation. The development of an efficient control system for a fluid flow should be based on the specific Navier-Stokes equations that describe the flow in order to exploit their ability to accurately predict the spatiotemporal behavior of the flow field. Navier-Stokes equations have motivated an extensive research activity on the dynamics where important contributions include the realization that turbulent flows involve coherent structures [21,78] and their computations, see [201,234–236]. The prediction of the behavior of Navier-Stokes equation is of great importance due to its wide application in the physical problems to model various phenomenon such as in theory of modeling of gas dynamics [37], traffic flow, investigating the shallow water waves [156], examining the chemical reaction-diffusion model [8], turbulent flow patterns of soil water [262], shock waves traveling in hydrodynamics turbulence [52].

The first approximate analytical solutions of time fractional Navier-Stokes equation were obtained by employing the Laplace transform method, finite Hankel transforms and finite Fourier sine transform by El-Shahed and Salem [60]. After this seminal work, the nonlinear fractional Navier - Stokes equation has been studied by many techniques such as adomaian decomposition method (ADM) [34, 177], Homotopy analysis method (HAM) [68, 200]. Recently, the fractional Navier-Stokes equation was solved by employing residual power series method [100] and a new method based on operational matrices [238].

One dimensional Burgers' equation: $\frac{\partial u}{\partial t} + \alpha u \frac{\partial u}{\partial x} - \nu \frac{\partial^2 u}{\partial x^2} = 0$, the simplest nonlinear partial differential equation for diffusive waves in fluid dynamics consists two terms: non-linear convection term and viscous diffusion term. Thus, it becomes a simplified form to a one-dimensional analogue of the Navier-Stokes equations without stress terms. First time, it was also used in the context of a statistical theory of turbulent motion of fluids [26] and later describing a mathematical model of turbulence [35–37], due to such extensive work of Burgers' this equation is said to be Burgers' equation. This equation arises in many physical problems including one-dimensional turbulence, shock/sound waves in a viscous

medium, waves in fluid-filled viscous elastic tubes, continuous stochastic processes, vorticity transportation, dispersion in porous media and magneto-hydrodynamic waves in a medium with finite electrical conductivity and many more. The more details, we refer the readers to [35–37, 52, 62, 65].

In the last years, a lot of efforts have been made to compute the accuracy and efficiency of various numerical schemes for Burgers' equation with various values of kinematic viscosity. Burgers' equation has already been solved using several analytical and numerical schemes, for, instance, Hofe Cole transformation [52, 65], finite element method [9], finite difference method [81, 260], implicit finite difference method [104, 261], compact finite difference method [30, 153, 174], fourier pseudospectral method [202], variational iteration method [6], reproducing kernel function method [227], quadratic B-spline finite elements [192], collocation methods based on cubic B-spline [55, 168], modified cubic B-splines [171] and extended B-spline [61]. The interested readers also read [13, 30].

The fractional model of one dimensional coupled viscous Burgers' equation has been solved by employing the reduced differential transform method [247, 251, 256], fractional reduced differential transform [237], generalized differential transform method [155], coupling of homotopy perturbation method and Padè technique [108], homotopy perturbation method [85], ADM [291], Laplace homotopy algorithm [243], Laplace homotopy perturbation method [103] and Fractional variational iteration method [193]. For more details of HPM, we refer the readers to [20, 187, 206, 286] and references therein.

Partial differential equations with proportional delays is a special class of delay partial differential equation, arises in the field of complex economic macro-dynamics [124], medicine, control systems and climate models [279]. A little literature was found for the implementation of numerical methods for the study of the solution behaviors of time fractional partial differential equations (TFPDEs) with delay. Some methods which were adopted for the study of classical/fractional differential equations with delay are reported in Chapter 7.

*Kuramoto-Sivashinsky* equation (1.12.1) was originally derived in the context of plasma instabilities, flame front propagation, and phase turbulence in the reaction-diffusion system

[198].

$$\frac{\partial u}{\partial t} = \beta \frac{\partial^2 u}{\partial x^2} - u \frac{\partial u}{\partial x} + \gamma \frac{\partial^4 u}{\partial x^4} \tag{1.12.1}$$

where $\mathbb{R}$ is the set of real numbers, $f, g, \psi_1, \psi_2$ are known functions and $u_x = \frac{\partial u}{\partial x}, u_{xx} = \frac{\partial^2 u}{\partial x^2}$, etc. The nonlinear term in KS equation counterbalances the dispersion term while dissipation terms show a mechanism for energy transfer. Moreover, KS equation with $\gamma = 0$ is the known Burgers' equation.

*Kuramoto-Sivashinsky* equation models the fluctuations of the position of a flame front, the motion of a fluid going down a vertical wall, or a spatially uniform oscillating chemical reaction in a homogeneous medium [54]. *Kuramoto-Sivashinsky* equation exhibits chaotic behavior, having solution like traveling waves moving without change of shape over a finite spatial domain. *Kuramoto-Sivashinsky* equation occurred in many physical phenomena: reaction diffusion systems [144], long waves on the interface between two viscous fluids [97], hydrodynamics thin films [266], and flame front instability [253].

In the past years, the *Kuramoto-Sivashinsky* equation was studied numerically by adopting rigorous approaches such as Chebyshev spectral collocation methods [113], local discontinuous Galerkin methods [284], tanh function method [150], homotopy analysis method [146], inverse scattering method [59], homogeneous balance method [64], cubic B-spline finite difference collocation method [149], quintic B-spline collocation method (QBSC) [167], septic B-spline collocation method (SBSC) [295], higher-order finite element approach [16], finite difference discretization [10], fourth-order singly diagonally implicit Runge-Kutta method [58], lattice Boltzmann method (LBM) [285]. The asymptotic states of the damped Kuramoto-Sivashinsky equations were studied numerically in [73].

## 1.13 Thesis Plan

In continuation of the introduction, the proposed work of the thesis is organized as follows In **Chapter 2**, a novel approach: *modified extended cubic B-spline differential quadrature (mECDQ) method* in space discretization has been developed with time integration algorithm for numerical simulation of initial values system of nonlinear Burgers' equation in

$(1+1)$ dimension and coupled Burgers equation in $(n+1)$ dimension $(n = 1, 2)$ with appropriate Boundary conditions. The mECDQ method (DQM with modified extended cubic B-splines as base functions) is used to convert the initial boundary value system of the Burgers' equation into an initial value system of ordinary differential equations (ODEs), in time. We prefer an optimal five stage four order strong stability preserving Runge-Kutta method (SSP-RK54) to solve this resulting system of ODEs. Six test problems are considered to test the accuracy and efficiency of mECDQ method. The proposed results are compared with the exact solutions in terms of $L_2$ and $L_\infty$ errors and the existing results. The mECDQ scheme is shown conditionally stable Burgers' equations.

**Chapter 3** concerns with a new method *modified trigonometric cubic B-spline differential quadrature method (MTB-DQM)* in space discretization with SSP-RK54 algorithm for solving the time-dependent PDEs. Specially, the proposed algorithm has been implemented for nonlinear Burgers' equations. First, MTB-DQM (DQM with modified trigonometric cubic B-splines as base functions) is used to convert the initial boundary value system of Burgers' equation into the initial value system of first-order ODEs, in time, after that SSP-RK54 algorithm has been employed for solving the resulting system of ODEs. Four test problems are considered to illustrate the accuracy/efficiency of the method in terms of $L_2$ and $L_\infty$ error norms and their comparisons with existing results. Moreover, MTB-DQM is shown conditionally stable for various grid points and computed presented results are better than the results obtained by almost all the existing schemes.

**Chapter 4** adopted a *compact finite difference scheme* (Lele, 1992) in the space discretizations while optimal four-stage, order three strong stability-preserving time-stepping Runge-Kutta scheme in the time, for the numerical simulation of one-dimensional Kuramoto-Sivashinsky equation "$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \beta\frac{\partial^2 u}{\partial x^2} + \gamma\frac{\partial^4 u}{\partial x^4}$", arises in the study of flame front propagation, phase turbulence in the reaction-diffusion system and in many other biological and chemical processes. The efficiency of the proposed scheme is confirmed by six test problems with known exact solutions. The numerical results demonstrate the reliability and efficiency of the algorithm developed.

**Chapter 5** deals with an analytical study of time-fractional Navier-Stokes equation,

considering the fractional derivative of Caputo type:

$$\mathcal{D}_t^{\alpha} U + (U \cdot \nabla)U = \nu \nabla^2 U - \tfrac{1}{\rho}\nabla p, \quad \text{on } \Omega \times (0, T)$$

The approximate analytical solutions are obtained by adopting two reliable methods: Fractional reduced differential transform method and a new integral projected differential transform method. The accuracy and efficiency of these methods are illustrated by three test problems of the time fractional Navier-Stokes equation. The scheme is found to be very reliable, effective and efficient powerful technique to solve a wide range of problems arising in engineering and sciences. The small size of computation FRDTM contrary to the other schemes is its strength.

**Chapter 6** deals with an approximate analytical solution of multi-dimensional, time-fractional coupled viscous Burgers' (TFCB) equation obtained by employing "homotopy perturbation method". The validity and efficiency of the homotopy perturbation method has been illustrated by considering three different examples of TFCB equation. The results are also depicted in graphically for different values of fractional order $\alpha$ and Reynolds number. It is found that the proposed series solutions converge rapidly for large Reynolds numbers (Re$\geq$ 100).

In **Chapter 7**, at first some properties of $(n+1)$-dimensional Extended FRDTM for delayed TFPDEs are presented. Approximate analytic solutions of $(1+1)$ dimensional TFPDEs with proportional delay and generalized Burgers' equations with proportional delay are obtained by two reliable methods: 1) fractional variation iteration method (FVIM), and 2) Extended fractional reduced differential transform method (Extended FRDTM). The approximate solutions from either method are obtained in a series form that converges to the exact solution behaviors very fast. The efficiency/validity of these methods is illustrated by three test problems of TFPDEs with proportional delay. The finding shows that Extended FRDTM is easy to implement as compared to FVIM. The small size of computation of Extended FRDTM is its strength.

In **Chapter 8**, the *homotopy perturbation transform method* (i.e., hybrid of homotopy perturbation technique & Laplace transform) has been implemented for solving initial

value autonomous system of time-fractional partial differential equations (TFPDEs) with proportional delay, including generalized Burgers' equations with proportional delay. The numerical study of three examples of TFPDEs with proportional delay is presented to test the efficiency and validity of proposed HPTM. The obtained solutions are in series form, converges very fast. The HPTM seems very reliable, effective and efficient powerful technique for study of many physical models arising in various branches of sciences and engineering.

# Chapter 2

# Modified extended cubic B-spline DQM for solving Burgers' Equation

## 2.1 Introduction

This chapter is concerned with nonlinear Burgers' equation (2.1.1) and coupled viscous Burgers' equation (2.1.2) in (1+1) dimension, and coupled viscous Burgers' equation (2.1.3) in $(2+1)$ dimension together with suitable initial and Dirichlet Boundary conditions:

$$\begin{aligned} &\frac{\partial u}{\partial t}+\alpha u\frac{\partial u}{\partial x}-\nu\frac{\partial^2 u}{\partial x^2}=0,\ u(x,0)=\psi(x),\ x\in\Omega_1,\ t>0,\\ &u(x,t)=\varsigma(x,t), x\in\partial\Omega_1, t\geq 0, \end{aligned} \tag{2.1.1}$$

$$\begin{aligned} &\frac{\partial u}{\partial t}=\frac{\partial^2 u}{\partial x^2}-\eta u\frac{\partial u}{\partial x}-\alpha\frac{\partial (uv)}{\partial x},\ u(x,0)=\phi(x),\\ &\frac{\partial u}{\partial t}=\frac{\partial^2 v}{\partial x^2}-\xi v\frac{\partial v}{\partial x}-\beta\frac{\partial (uv)}{\partial x},\ v(x,0)=\psi(x)\ x\in\Omega_1,\ t>0.\\ &u(a,t)=g_1(t),\ u(b,t)=g_2(t),\ v(a,t)=g_3(t),\ v(b,t)=g_4(t), t>0, \end{aligned} \tag{2.1.2}$$

$$\begin{aligned} &\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=\nu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right),\ u(x,y,0)=\psi_1(x,y),\\ &\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=\nu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right),\ v(x,y,0)=\psi_2(x,y),\ (x,y)\in\Omega, t>0\\ &u(x,y,t)=\xi(x,y,t),\ v(x,y,t)=\zeta(x,y,t), (x,y)\in\partial\Omega,\ t\geq 0, \end{aligned} \tag{2.1.3}$$

where $\Omega_1 = [a, b]$, $g_\ell(\ell = 1, 2, 3, 4), \phi, \psi$ are known smooth functions, the parameters $\xi, \eta$ real constants, $\alpha, \beta$ be the other constants depend on the system parameters (e.g., Peclet number, stokes velocity of particles and Brownian diffusivity [182]). $\Omega = [a, b] \times [c, d] \rightarrow$ computational domain, and its boundary is $\partial\Omega$, $u, v \rightarrow$ velocity components, $\frac{\partial u}{\partial t} \rightarrow$ unsteady term, $u\frac{\partial u}{\partial x} \rightarrow$ nonlinear convection term, $\nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \rightarrow$ diffusion term and $\nu \rightarrow$ coefficient of viscosity ($\frac{1}{Re} = \nu > 0$), $Re \rightarrow$ Reynolds number and $\alpha > 0$, a constant, and $\psi, \psi_1, \psi_2, \xi$ and $\zeta$ are known functions.

Bellman et al. [27] developed the differential quadrature (DQ) method for solving partial differential equations (PDEs). After the seminal paper of Bellman et al., and Quan and Chang [196, 197] the differential quadrature method has been implemented for various type of a set of base functions, among others, cubic B-spline DQ methods [124, 128, 129], DQM based on Fourier expansion and Harmonic function [226, 228], Polynomial based DQM [124,131], quartic B-spline based DQM [24,133], Quartic and quintic B-spline methods [130], exponential cubic B-spline DQM [126], sinc DQM [134], generalized DQM [223], exponential cubic B-spline based DQM [126] and modified cubic B-spline DQ method (MCB-DQM) [13, 239–241]. Having capability to handle local phenomena, B-splines (piece-wise smooth polynomials) have more influence in comparison to another set of basis functions. A lot of physical models have already been solved by considering the (modified) cubic B-splines as a set of base functions. Recently, Korkmaz and Akmaz have developed extended cubic B-spline DQM [127] which produces stable solutions for one dimensional Burgers' equation.

In this chapter, a modified extended cubic-B-spline differential quadrature (mECDQ) method has been developed for the numerical computation of the Burgers' equation. The mECDQ method is the DQM with modified extended cubic-B-splines as a new set of base functions. The mECDQ method converts the initial- boundary value system of Burgers' equation into an initial value system of ODEs in time, which we solve by an optimal five - stage, order four strong stability-preserving time-stepping Runge-Kutta (SSP-RK54) algorithm. The accuracy, efficiency, and adaptability of the method is confirmed by taking six test problems.

## 2.2 Description of the method

This section deals with the description of mECDQ method for Burgers' equation. Being the weighting coefficients depend on grid spacing only, the domains $\Omega$ and $\Omega_1$ defined by $\Omega_1 = \{x \in \mathbb{R} : a \leq x \leq b\}$ *and* $\Omega = \{(x, y) \in \mathbb{R}^2 : a \leq x \leq b, \quad c \leq y \leq d\}$ are portioned uniformly in each direction with the following knots:

$$\begin{aligned} &a = x_1 < x_2, \ldots < x_i < \ldots < x_{N_x - 1} < x_{N_x} = b, \\ &c = y_1 < y_2, \ldots < y_j < \ldots < y_{N_y - 1} < x_{N_y} = d, \end{aligned} \tag{2.2.1}$$

where $h_x = \frac{b-a}{N_x - 1}$; $h_y = \frac{d-c}{N_y - 1}$, is the descritization step in $x$ and $y$ directions, respectively.

Let $x_i \in \Omega_1, (x_i, y_j) \in \Omega$ be generic grid points and

$$u_i = u_i(t) = u(x_i, t); \quad u_{ij} = u_{ij}(t) = u(x_i, y_j, t)$$

The $r$-th order partial derivatives of $u(x, t)$ with respect to $x$ approximated at $x_i$ read:

$$\frac{\partial^r u_i}{\partial x^r} = \sum_{k=1}^{N_x} a_{ik}^{(r)} u_k, \quad i \in \Delta_{N_x} \tag{2.2.2}$$

The $r$-th order partial derivatives of $u(x, y, t)$ with respect to $x, y$ at the grid point $(x_i, y_j)$ can be computed as follows:

$$\begin{cases} \dfrac{\partial^r u}{\partial x^r}(x_i, y_j) = \displaystyle\sum_{k=1}^{N_x} a_{ik}^{(r)} u_{kj} \quad i \in \Delta_{N_x} \\ \dfrac{\partial^r u}{\partial y^r}(x_i, y_j) = \displaystyle\sum_{k=1}^{N_y} b_{jk}^{(r)} u_{ik} \quad j \in \Delta_{N_y} \\ \dfrac{\partial^r v}{\partial x^r}(x_i, y_j) = \displaystyle\sum_{k=1}^{N_x} a_{ik}^{(r)} v_{kj} \quad i \in \Delta_{N_x} \\ \dfrac{\partial^r v}{\partial y^r}(x_i, y_j) = \displaystyle\sum_{k=1}^{N_y} b_{jk}^{(r)} v_{ik} \quad j \in \Delta_{N_y} \end{cases} \tag{2.2.3}$$

where $a_{ij}^{(r)}$ and $b_{ij}^{(r)}$ $(r = 1, 2)$ are the weighting coefficients of the $r$-th order partial derivatives with respect to $x$ and $y$.

The extended cubic B-spline basis function is defined as [127]:

$$\varphi_i = \frac{1}{24}\begin{cases} 4(1-\lambda)P_{i-2}^3(x) + 3\lambda P_{i-2}^4(x), & x \in [x_{i-2}, x_{i-1}) \\ 24p + 12P_{i-1}(x) + 6(2+\lambda)P_{i-1}^2(x) - 12P_{i-1}^3(x) - 3\lambda P_{i-1}^4(x), & x \in [x_{i-1}, x_i) \\ 24p - 12P_{i+1}(x) + 6(2+\lambda)P_{i+1}^2(x) + 12P_{i+1}^3(x) - 3\lambda P_{i+1}^4(x), & x \in [x_i, x_{i+1}) \\ 4(\lambda-1)P_{i+2}^3(x) + 3\lambda P_{i+2}^4(x), & x \in [x_{i+1}, x_{i+2}) \\ 0, & \text{otherwise} \end{cases}$$

where $h_x P_i(x) = (x - x_i)$ and $24\wp = 4 - \lambda$, and $\lambda$ is a free parameter, used to obtain different forms of the extended cubic B-splines. For instance, extended cubic B-splines with $\lambda = 0$ are the cubic B-splines. The set of extended cubic B-splines $\{\varphi_0, \varphi_1, \varphi_2, \ldots, \varphi_{N_x}, \varphi_{N_x+1}\}$ forms a basis over $\Omega_1$. Let $12h\theta = 8 + \lambda$ and $2h^2\omega = 2 + \lambda$. The values of extended cubic B-spline $\varphi_{ij} = \varphi_i(x_j)$ and its first and second derivatives at $x_j$, is $\varphi_{ij} := \varphi_i(x_j)$, $\varphi'_{ij} := \varphi'_i(x_j)$ and $\varphi''_{ij} := \varphi''_i(x_j)$, respectively, read as:

$$\varphi_{ij} = \begin{cases} \theta, \text{if } i-j=0 \\ \wp, \text{if } i-j=\pm 1 \\ 0, \text{if } otherwise \end{cases} ; \ \varphi'_{ij} = \begin{cases} \frac{1}{2h_x}, \text{if } i-j=1 \\ -\frac{1}{2h_x}, \text{if } i-j=-1 \\ 0, \text{if } otherwise \end{cases} ; \ \varphi''_{ij} = \begin{cases} -2\omega, \text{if } i-j=0 \\ \omega, \text{if } i-j=\pm 1 \\ 0, \text{if } otherwise \end{cases} \quad (2.2.4)$$

Taken idea from [13], the extended cubic B-splines are modified as:

$$\begin{cases} \phi_1(x) = \varphi_1(x) + 2\varphi_0(x) \\ \phi_2(x) = \varphi_2(x) - \varphi_0(x) \\ \vdots \\ \phi_j(x) = \varphi_j(x), \text{ for } j = 3, 4, \ldots, N_x - 2 \\ \vdots \\ \phi_{N_x-1}(x) = \varphi_{N_x-1}(x) - \varphi_{N_x+1}(x) \\ \phi_{N_x}(x) = \varphi_{N_x}(x) + 2\varphi_{N_x+1}(x) \end{cases} \quad (2.2.5)$$

The set $\{\phi_1, \phi_2, \ldots, \phi_{N_x}\}$ forms a basis of the domain $\Omega_1$.

## 2.3 Computation of the weighting coefficients

In order to evaluate the weighting coefficients $a_{ij}^{(1)}$ of Eq. (2.2.3), the modified extended cubic B-splines $\phi_m(x), m \in \Delta_{N_x}$ are used. Taking $\phi_{mi}^{'} := \phi_m^{'}(x_i), \phi_{m_i} := \phi_m(x_i)$. Accordingly, approximation of the first order spatial derivative is given as follows:

$$\phi_{mi}^{'} = \sum_{\ell=1}^{N_x} a_{i\ell}^{(1)} \phi_{m\ell} \tag{2.3.1}$$

Setting $\Phi = [\phi_{ml}],\ A = [a_{i\ell}^{(1)}]$(the unknown weighting coefficient matrix), and $\Phi^{'} = [\phi_{mi}^{'}]$ then Eq. (2.3.1) can be written as the set of a tridiagonal system of linear equations:

$$\Phi A^T = \Phi^{'} \tag{2.3.2}$$

The coefficient matrix $\Phi$ of order $N_x$ can be computed from Eqs. (2.2.4) and (2.2.5) as:

$$\Phi = \begin{bmatrix} 1 & \wp & & & & & \\ 0 & \theta & \wp & & & & \\ & \wp & \theta & \wp & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \wp & \theta & \wp & \\ & & & & \wp & \theta & 0 \\ & & & & & \wp & 1 \end{bmatrix}$$

and in particular the columns of the matrix $\Phi'$ read:

$$\Phi'[1] = \begin{bmatrix} -1/h \\ 1/h \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \Phi'[2] = \begin{bmatrix} -1/2h \\ 0 \\ 1/2h \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \ldots, \Phi'[N_x-1] = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -1/2h \\ 0 \\ 1/2h \end{bmatrix}, \Phi'[N_x] = \begin{bmatrix} 0 \\ \\ \vdots \\ 0 \\ -1/h \\ 1/h \end{bmatrix}.$$

Being $\Phi$ invertible, the tridiagonal system Eq. (2.3.2) is solved by using Thomas algorithm [151] for the weighting coefficients $a_{i\ell}^{(1)}, i, \ell \in \Delta_{N_x}$ Similarly, the weighting coefficients $b_{i\ell}^{(1)}$ can be computed by considering the grid points in $y$-direction.

One can determine $a_{i\ell}^{(2)}, \quad i, \ell \in \Delta_{N_x}$ as in [128]: $\phi''_{mi} = \sum_{\ell=1}^{N_x} a_{i\ell}^{(2)} \phi_{m\ell}, \ i, m \in \Delta_{N_x}$. Similar to system Eq.(2.3.2), this system yields $a_{i\ell}^{(2)} \quad i, \ell \in \Delta_{N_x}$. The existence of more than one basis functions to span the $N_x$-dimensional computational domain gives an opportunity to determine the weighting coefficients $a_{i\ell}^{(r)}, \ i, \ell \in \Delta_{N_x}$ in the same space by means of various set of basis functions. Keeping the computation cost in mind, we prefer Shu's $rth$-order ($r \geq 2$) recursive formula [223, 225] based on polynomial based DQM, to compute the weighting coefficients $a_{i\ell}^{(2)}, i, \ell \in \Delta_{N_x}; b_{i\ell}^{(2)}, i, \ell \in \Delta_{N_y}$ as follows:

$$\begin{cases} a_{i\ell}^{(r)} = r\left(a_{i\ell}^{(1)} a_{ii}^{(r-1)} - \dfrac{a_{i\ell}^{(r-1)}}{x_i - x_\ell}\right), & \mathrm{i} \neq \ell, \mathrm{i}, \ell \in \Delta_{N_x}; \\ a_{ii}^{(r)} = -\displaystyle\sum_{\ell=1, \ell \neq i}^{N_x} a_{i\ell}^{(r)}, & \mathrm{i} = \ell, \mathrm{i}, \ell \in \Delta_{N_x}; \\ b_{i\ell}^{(r)} = r\left(b_{i\ell}^{(1)} b_{ii}^{(r-1)} - \dfrac{b_{i\ell}^{(r-1)}}{y_i - y_\ell}\right), & \mathrm{i} \neq \ell, \mathrm{i}, \ell \in \Delta_{N_y}; \\ b_{ii}^{(r)} = -\displaystyle\sum_{\ell=1, \ell \neq i}^{N_y} b_{i\ell}^{(r)}, & \mathrm{i} = \ell, \mathrm{i}, \ell \in \Delta_{N_y}; \end{cases} \tag{2.3.3}$$

### 2.3.1 Implementation of mECDQ method to Burgers' equations in $(1+1)$ and $(2+1)$ dimensions

#### A. Burgers' and coupled Burgers' equations in $(1+1)$ dimension

On putting the values of the spatial derivatives approximated by mECDQ method, Eq.(2.1.1) with initial condition can be re-written as follows:

$$\begin{cases} \dfrac{\partial u}{\partial t}(x_i) = -\alpha u_i \displaystyle\sum_{j=1}^{N_x} a_{ij}^{(1)} u_j - \nu \sum_{j=1}^{N_x} a_{ij}^{(2)} u_j, \mathrm{i} \in \Delta_{N_x} \\ u_i(t=0) = \psi_i, \qquad\qquad i \in \Delta_x \end{cases} \tag{2.3.4}$$

Considering boundary conditions in mind, Eq. (2.3.4) reduces to a set of first order ODEs:

$$\begin{cases} \dfrac{\partial u_i}{\partial t} = \nu \sum_{j=2}^{N_x-1} a_{ij}^{(2)} u_j - \alpha_i \sum_{j=2}^{N_x-1} a_{ij}^{(1)} u_j + F_i & i \in \Delta_{N_x} \\ u_i(t=0) = \psi_i, & i \in \Delta_{N_x} \end{cases} \tag{2.3.5}$$

where

$$F_i = \nu \left( a_{i1}^{(2)} u_1 + a_{iN_x}^{(2)} u_{N_x} \right) - \alpha_i \left( a_{i1}^{(1)} u_1 + a_{iN_x}^{(1)} u_{N_x} \right), \tag{2.3.6}$$

Now, the one dimensional coupled viscous Burgers' equation (2.1.2) can be re-written as:

$$\begin{cases} \dfrac{\partial u_i}{\partial t} = \sum_{j=1}^{N_x} a_{ij}^{(2)} u_j - \eta u_i \sum_{j=1}^{N_x} a_{ij}^{(1)} u_j - \alpha \left( u_i \sum_{j=1}^{N_x} a_{ij}^{(1)} v_j + v_i \sum_{j=1}^{N_x} a_{ij}^{(1)} u_j \right) \\ \dfrac{\partial v_i}{\partial t} = \sum_{j=1}^{N_x} a_{ij}^{(2)} v_j - \xi v_i \sum_{j=1}^{N_x} a_{ij}^{(1)} v_j - \beta \left( u_i \sum_{j=1}^{N_x} a_{ij}^{(1)} v_j + v_i \sum_{j=1}^{N_x} a_{ij}^{(1)} u_j \right) \\ u_i(0) = \phi(x_i),\ v_i(0) = \psi(x_i),\ i \in \Delta_{N_x} \end{cases} \tag{2.3.7}$$

On implementing the boundary conditions from (2.1.2), system (2.3.8) reduces to a set of first order ODEs of the form:

$$\begin{cases} \dfrac{\partial u_i}{\partial t} = \sum_{j=1}^{N_x-1} a_{ij}^{(2)} u_j - \eta u_i \sum_{j=1}^{N_x-1} a_{ij}^{(1)} u_j - \alpha \left( u_i \sum_{j=1}^{N_x-1} a_{ij}^{(1)} v_j + v_i \sum_{j=1}^{N_x-1} a_{ij}^{(1)} u_j \right) + F_i \\ \dfrac{\partial v_i}{\partial t} = \sum_{j=1}^{N_x-1} a_{ij}^{(2)} v_j - \xi v_i \sum_{j=1}^{N_x-1} a_{ij}^{(1)} v_j - \beta \left( u_i \sum_{j=1}^{N_x-1} a_{ij}^{(1)} v_j + v_i \sum_{j=1}^{N_x-1} a_{ij}^{(1)} u_j \right) + G_i \\ u_i(0) = \phi(x_i),\ v_i(0) = \psi(x_i),\ 1 < i \in < N_x \end{cases} \tag{2.3.8}$$

where $\eta_i = \eta u_i, \xi_i = \xi v_i, \alpha_i = \alpha u_i, \alpha_i' = \alpha v_i, \beta_i = \beta u_i, \beta_i' = \beta v_i$, and

$$F_i = (a_{i1}^{(2)} u_1 + a_{iN_x}^{(2)} u_{N_x}) - \eta_i (a_{i1}^{(1)} u_1 + a_{iN_x}^{(1)} u_{N_x}) - \alpha_i (a_{i1}^{(1)} v_1 + a_{iN_x}^{(1)} v_{N_x}) - \alpha_i' (a_{i1}^{(1)} u_1 + a_{iN_x}^{(1)} u_{N_x}),$$

$$G_i = (a_{i1}^{(2)} v_1 + a_{iN_x}^{(2)} v_{N_x}) - \xi_i (a_{i1}^{(1)} v_1 + a_{iN_x}^{(1)} v_{N_x}) - \beta_i (a_{i1}^{(1)} v_1 + a_{iN_x}^{(1)} v_{N_x}) - \beta_i' (a_{i1}^{(1)} u_1 + a_{iN_x}^{(1)} u_{N_x}).$$

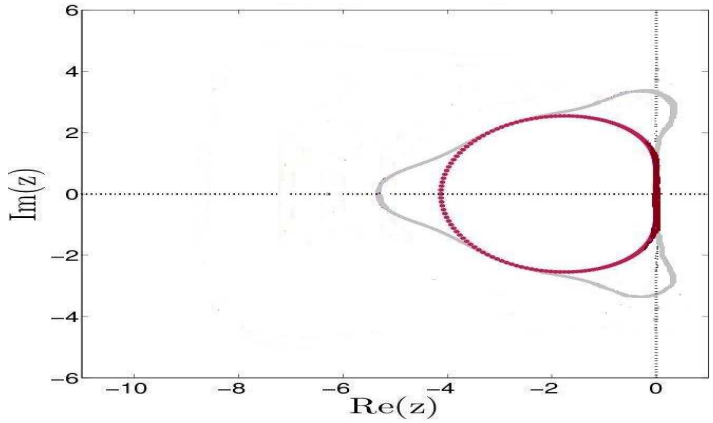


Figure 2.1: Stability region of SSP-RK54 algorithm with $z = \lambda\Delta t$

**B. Nonlinear coupled Burgers' equations in $(2+1)$ dimension**

Similarly, on implementing mECDQ method in space, Eqs. (2.1.3) reduces to

$$\begin{cases} \dfrac{\partial u_{ij}}{\partial t} = \nu\left(\sum\limits_{k=2}^{N_x-1} a_{ik}^{(2)} u_{kj} + \sum\limits_{k=2}^{N_y-1} b_{jk}^{(2)} u_{ik}\right) - u_{ij}\sum\limits_{k=2}^{N_x-1} a_{ik}^{(1)} u_{kj} - v_{ij}\sum\limits_{k=2}^{N_y-1} b_{jk}^{(1)} u_{ik} + F_{ij}, \\ \dfrac{\partial v_{ij}}{\partial t} = \nu\left(\sum\limits_{k=2}^{N_x-1} a_{ik}^{(2)} v_{kj} + \sum\limits_{k=2}^{N_y-1} b_{jk}^{(2)} v_{ik}\right) - u_{ij}\sum\limits_{k=2}^{N_x-1} a_{ik}^{(1)} v_{kj} - v_{ij}\sum\limits_{k=2}^{N_y-1} b_{jk}^{(1)} v_{ik} + G_{ij}, \\ u_{(ij)}(t=0) = \xi(x_i, x_j), v_{ij}(t=0) = \zeta(x_i, x_j), 1 < i < \Delta_{N_x}, 1 < j < \Delta_{N_y} \end{cases} \tag{2.3.9}$$

where $u_{ij} = \tau_{ij}$ and $v_{ij} = \kappa_{ij}$ and

$$\begin{aligned} F_{ij} &= \nu(a_{i1}^{(2)} u_{1j} + a_{iN_x}^{(2)} u_{N_xj} + b_{j1}^{(2)} u_{i1} + b_{jN_y}^{(2)} u_{iN_y}) - \tau_{ij}(a_{i1}^{(1)} u_{1j} + a_{iN_x}^{(1)} u_{N_xj}) \\ &\quad - \kappa_{ij}(b_{jl}^{(1)} u_{il} + b_{jN_y}^{(1)} u_{iN_y}), \\ G_{ij} &= \nu(a_{i1}^{(2)} v_{1j} + a_{iN_x}^{(2)} v_{N_xj} + b_{j1}^{(2)} v_{i1} + b_{jN_y}^{(2)} v_{iN_y}) - \tau_{ij}(a_{i1}^{(1)} v_{1j} + a_{iN_x}^{(1)} v_{N_xj}) \\ &\quad - \kappa_{ij}(b_{jl}^{(1)} v_{il} + b_{jN_y}^{(1)} v_{iN_y}). \end{aligned} \tag{2.3.10}$$

The initial value system (2.3.5) and system (2.3.9) of first order ODEs can be solved via many time integration algorithms, among others, SSP-RK algorithms allow low storage and large domain of absolute properties [77, 135, 254] which results in less accumulation of the numerical errors. In particular, we prefer SSP-RK54 algorithm [77, 254] for solving of

these initial value systems through the following steps [135, 254]:

$$
\begin{aligned}
U^{(1)} &= U^m + 0.391752226571890\triangle tL(U^m)\\
U^{(2)} &= 0.444370493651235U^m + 0.555629506348765U^{(1)} + 0.368410593050371\triangle tL(U^{(1)})\\
U^{(3)} &= 0.620101851488403U^m + 0.379898148511597U^{(2)} + 0.251891774271694\triangle tL(U^{(2)})\\
U^{(4)} &= 0.178079954393132U^m + 0.821920045606868U^{(3)} + 0.544974750228521\triangle tL(U^{(3)})\\
U^{m+1} &= 0.517231671970585U^{(2)} + 0.096059710526147U^{(3)} + 0.063692468666290\triangle tL(U^{(3)})\\
&\quad + 0.386708617503269U^{(4)} + 0.226007483236906\triangle tL(U^{(4)}),
\end{aligned}
\tag{2.3.11}
$$

## 2.4 Stability analysis

In this section, the stability analysis of mECDQ method for Burgers' equation is studied. As in [209], Terms $\alpha_i = \alpha u_i$ in the nonlinear terms of Eq. (2.1.1), and $u_{ij} = \tau_{ij}$ and $v_{ij} = \kappa_{ij}$ in the nonlinear terms of Eq. (2.1.3) are assumed to be locally fixed. Let $A_2 = [a_{ij}^{(2)}]; A_1 = [a_{ij}^{(1)}]$ and $B_2 = [b_{ij}^{(2)}]; B_1 = [b_{ij}^{(1)}]$ be the matrices of the weighting coefficients of order $(N_x - 2)$. This reduces Eq. (2.3.5) to

$$\frac{dU}{dt} = AU + F,\ U(t=0) = U_0 \tag{2.4.1}$$

where $U = (u_2, u_3, \ldots, u_{N_x-1})$ and

$$A = \nu A_2 - \alpha_i A_1 \tag{2.4.2}$$

Analogously, Eq.(2.3.9) can be written as

$$\frac{dU}{dt} = BU + H, \tag{2.4.3}$$

where $\star$) $U = (u, v)^T$ is an unknown vector of the functional values at interior grid points:
$u = \left(u_{22}, \ldots, u_{2(N_y-1)}, u_{32}, \ldots, u_{3(N_y-1)} \ldots u_{(N_x-1)2}, u_{(N_x-1)3}, \ldots, u_{(N_x-1)(N_y-1)}\right)$ and
$v = \left(v_{22}, \ldots, v_{2(N_y-1)}, v_{32}, \ldots, v_{3(N_y-1)} \ldots v_{(N_x-1)2}, v_{(N_x-1)3}, \ldots, v_{(N_x-1)(N_y-1)}\right)$,

$\star$) $H = (F, G)^T, F = [F_{ij}], G = [G_{ij}] 1 < i < N_x; 1 < j < N_y$ as defined in Eq. (2.3.10),

$\star$)

$$B = \begin{bmatrix} A & O \\ O & A \end{bmatrix}, \text{ and } \quad A = -\tau_{ij} A_1 - \kappa_{ij} B_1 + \nu A_2 + \nu B_2, \tag{2.4.4}$$

where $O$'s null matrices, $A_r$ and $B_r (r = 1, 2)$, below are square block diagonal matrices of order $(N_x - 2)(N_y - 2)$ of the weighting coefficients $a_{ij}^{(r)}, b_{ij}^{(r)}$, respectively.

$$A_r = [w_{ij}^{(r)} I], \quad B_r = \begin{bmatrix} C_r & O_1 & \dots & O_1 \\ O_1 & C_r & \dots & O_1 \\ \vdots & \vdots & \ddots & \vdots \\ O_1 & O_1 & \dots & C_r \end{bmatrix}, C_r = [c_{ij}^{(r)}] \tag{2.4.5}$$

$\star$) $w_{ij}^{(r)} = a_{(i+1)(j+1)}^{(r)}$, $1 < i, j \in \Delta_{N_x - 2}$ and $c_{ij}^{(r)} = b_{(i+1)(j+1)}^{(r)}, \in \Delta_{N_y - 2}$. $I$ and $O_1$ are the matrices of order $N_y - 2$ and $N_x - 2$, respectively.

The stability of system (2.4.1) and system (2.4.3) depends on the eigenvalues of the matrices $A$ and $B$, respectively [101]. If the solution of the system (2.4.1) is decreasing in absolute value, then the system is stable whenever each eigenvalue of $A$ must have negative real part. In fact, the stability region is the set $S = \{z \in C, |R(z)| \leq 1, z = \lambda \triangle t\}$ where $R(.)$ is the stability function and $\lambda$ be the eigenvalue of the matrix $A$. The stability region of SSP-RK54 algorithm is depicted in Figure 2.1, see [135, Fig. 5].

Thus, the sufficient for the stability of system (2.4.1) is that $\lambda_A \Delta t$ lie inside the stability region $S$ (let) of SSP-RK54 algorithm for each eigenvalue $\lambda_A$ of $A$. Similarly, system (2.4.3) is stable whenever $\lambda_B \triangle t \in S$ for each eigenvalue $\lambda_B$ of $B$. For more details, see [101, 135].

It has been checked that the eigenvalues $\lambda_1$ and $\lambda_2$ of $A_1$ and $A_2$ in one dimension have the same nature as in two-dimensions. Moreover, the eigenvalues of $A_r$ and $B_r (r = 1, 2)$ are same, and so, it is sufficient to compute the eigenvalues $\lambda_1$ and $\lambda_2$. The values of $\lambda_1$, $\lambda_2$ for different step size $h$ are depicted in Figure 2.2, which shows that each eigenvalue $\lambda_1$ of $A_1$ is pure imaginary whereas each eigenvalue $\lambda_2$ of $A_2$ is real and negative. It is direct from Figure 2.2 and Eq. (2.4.2) that for given values of $h$ and $\nu$ one can find the time step $\triangle t$ for which $\lambda_A \triangle t$ lie inside $S$, the stability region of SSP-RK54 (see, Figure 2.1)

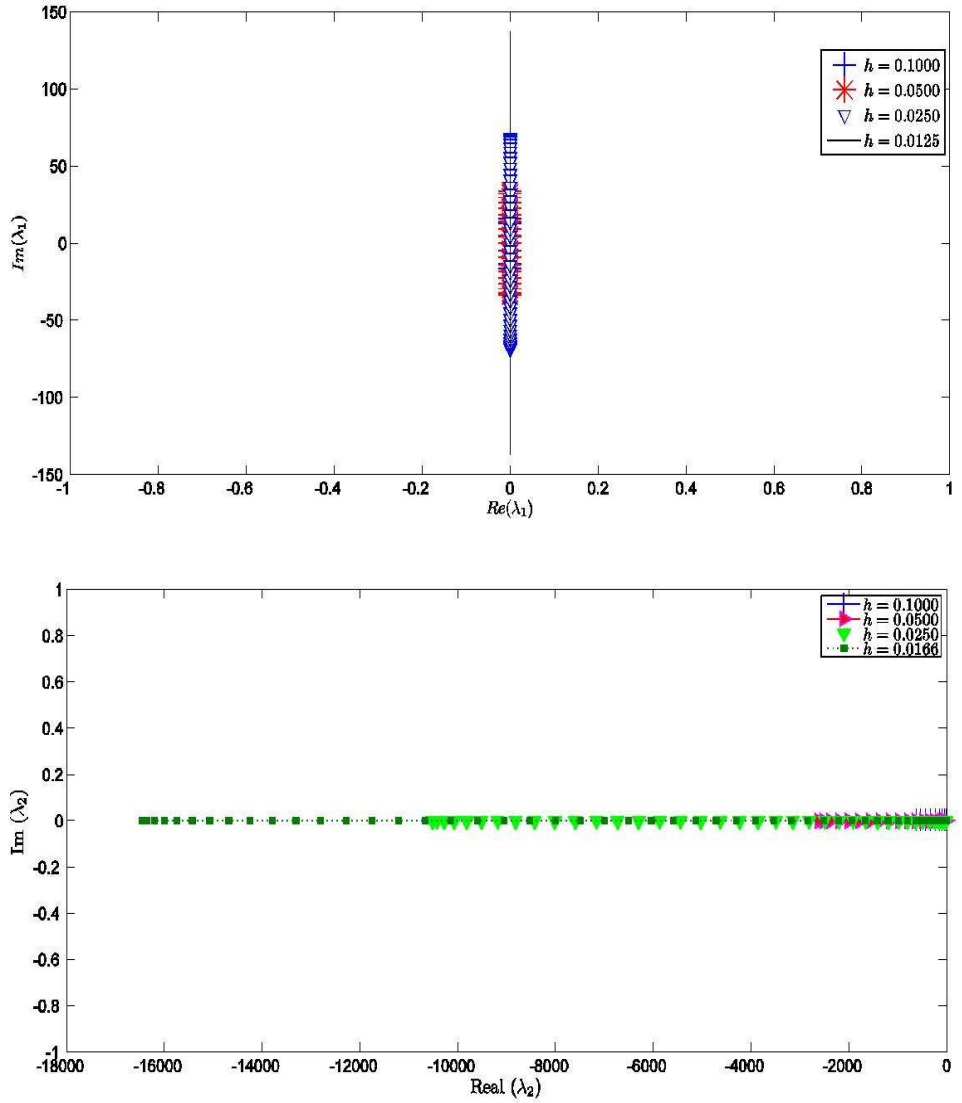


Figure 2.2: Eigenvalues $\lambda_1$ and $\lambda_2$ for different grid size $h$.

for each eigenvalue $\lambda_A = v\lambda_2 - \alpha_i\lambda_1$ of $A$. It is evident from Figure2.2 and Eq. (2.4.3) that for a given value of $h$ and $\nu$ $\exists$ $\triangle t$ for which $\lambda_B \triangle t$ corresponding to each eigenvalue $\lambda_B = 2v\lambda_2 - \lambda_1(\tau_0 + \kappa_0)$ of $B$ lies inside $S$. The above findings confirm that mECDQ method produces stable solutions for Burgers' equations in one and two dimensions.

In Example 2.5.1, for $h = 0.01$ the eigenvalues are computed for free parameter $\lambda = -0.012, \nu = 0.005$. It is found that each eigenvalue $\lambda_1$ belongs to $[-171.39i, 173.39i]$ whereas each $\lambda_2$ belongs to $[-65353, 0)$, and so, for any time step $\triangle t \leq 0.01$, the value $\lambda_A \triangle t = (v\lambda_2 - \alpha_i\lambda_1)\triangle t$ lies inside the stability region of SSP-RK54 algorithm. Similarly, the time step $\triangle t$ can be determined for producing table results in the other examples.

## 2.5 Numerical results and discussion

This section is devoted to the accuracy analysis of the proposed numerical method. Specifically two test case of (2.1.1) and three test cases of (2.1.3) are taken into account. The accuracy and consistency of the method is performed by considering the following $L_2$ and $L\infty$ error norms:

$$L_2 := \sqrt{h\sum_{j=1}^{n}|u_j - u_j^\star|^2}; \quad L_\infty := \max_j |u_j - u_j^\star| \tag{2.5.1}$$

where $u_j$ and $u_j^\star$ denote exact solution and computed solution at node $x_j$, respectively.

The efficiency of the method is performed by considering the rate of convergence (ROC) of the scheme, computed as follows:

$$ROC = \frac{\log_{10}\left(E(N_1)/E(N_2)\right)}{\log_{10}(N_2/N_1)},$$

where $E(N_i)$ $(i = 1, 2)$ denote the $L_2$ or $L_\infty$ error norm with $N_i$ grid points. These values are computed by using by C++.

### 2.5.1 $(1+1)$D nonlinear Burgers' and coupled Burgers' equation

**Example 2.5.1.** *The first test case deals with* $(1+1)D$ *Burger's equation* (2.1.1) *with* $\alpha = 1$ *for* $\Omega_1 = [1, 1.2]$ *as given in [13] with* $u(x,1) = \frac{x}{1+\exp\left(\frac{1}{4\nu}(x^2-\frac{1}{4})\right)}$ *with* $u(0,t) = 0$, $u(1.2,t) = 0$, *for* $t > 1$. *In this problem the initial condition is taken at* $t = 1$. *The exact solution for the problem is given by*

$$u(x,t) = \frac{\frac{x}{t}}{1 + (\frac{t}{t_0})^{1/2}\exp(x^2/4\nu t)}, \quad t \geq 1; \quad t_0 = \exp(1/8\nu) \tag{2.5.2}$$

*The numerical computation of Example 2.5.1 is performed for different values of* $t$, $1.7 \leq t \leq 3.5$ *for* $\triangle t = 0.01, h = 0.01$ . *In order to find the best value of* $\lambda$ *over the interval* $[-1, 1]$, *maximum absolute error at* $t = 2.5$ *is depicted in Figure 2.3 which confirms that the optimal value is* $\lambda = -0.012$. *Table 2.1 shows that the CPU time for mECDQ method is almost same as MCB-DQM [13] while the mECDQ solutions are comparatively more accurate. The comparison of mECDQ results, in terms of* $L_2$ *and* $L_\infty$ *error norms with the results*

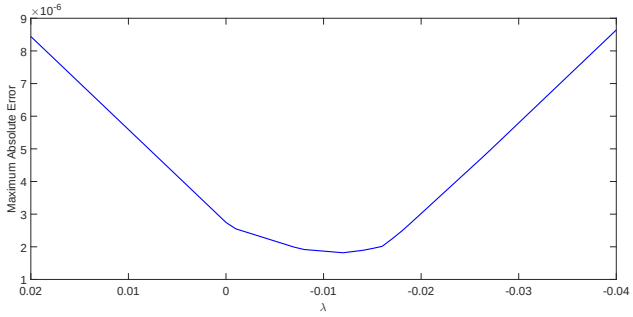


Figure 2.3: Behavior of $L_\infty$ error norm with respect to the free parameter $\lambda$.

Table 2.1: Comparison of computational cost of mECDQ of Example 2.5.1 with MCB-DQM [13] at $N_x = 121, t \leq 3.1, \triangle t = 0.01, \lambda = -0.012$ and $\nu = 0.005$

| | t=1.7 | | | t=2.5 | | | t=3.1 | | |
|---|---|---|---|---|---|---|---|---|---|
| | $L_2$ | $L_\infty$ | CPU | $L_2$ | $L_\infty$ | CPU | $L_2$ | $L_\infty$ | CPU |
| | $\times 10^6$ | | | $\times 10^6$ | | | $\times 10^6$ | | |
| MCB-DQM | 1.910 | 7.700 | 0.038 | 0.8600 | 3.080 | 0.089 | 0.6500 | 3.310 | 0.121 |
| mECDQ | 1.125 | 5.255 | 0.039 | 0.5182 | 2.043 | 0.075 | 0.6235 | 3.561 | 0.112 |

Table 2.2: Comparison of $L_2$ and$L_\infty$errors in the mECDQ solutions of Example 2.5.1 for $\nu = 0.005$ at $t = 3.6$

| | mECDQ | MCB-DQM | Korkmz and Dag [128] | | |
|---|---|---|---|---|---|
| | | | Method I | Method II | Method III |
| $L_2$ | 0.00001 | 0.00001 | 0.00018 | 0.00016 | 0.00014 |
| $L_\infty$ | 0.00007 | 0.00007 | 0.00046 | 0.50002 | 0.00054 |

*due to existing schemes in the literature, is reported in Table 2.2. The $L_2$ and $L_\infty$ error norms in mECDQ solutions are compared with the errors due to the schemes in [13, 128], is reported in Table 2.3. The above findings show that mECDQ results are much better in comparison to the results due to almost all earlier methods. The proposed method is the generalization of modified cubic B-spline differential quadrature method (i.e., mECDQ with free parameter $\lambda = 0$ reduces to modified cubic B-spline differential quadrature method). In mECDQ method one has the choice to optimize the results by considering the appropriate value of $\lambda$. Physical behavior of the mECDQ solutions for $\nu = 0.005$ at different time levels $t \leq 3.5$ with $h = 0.01, \triangle t = 0.01$ is depicted in Figure 2.4. The absolute errors for different time levels are depicted in Figure 2.5.*

**Example 2.5.2.** *The second test case deals with $(1+1)D$ coupled viscous equation* (2.1.2)

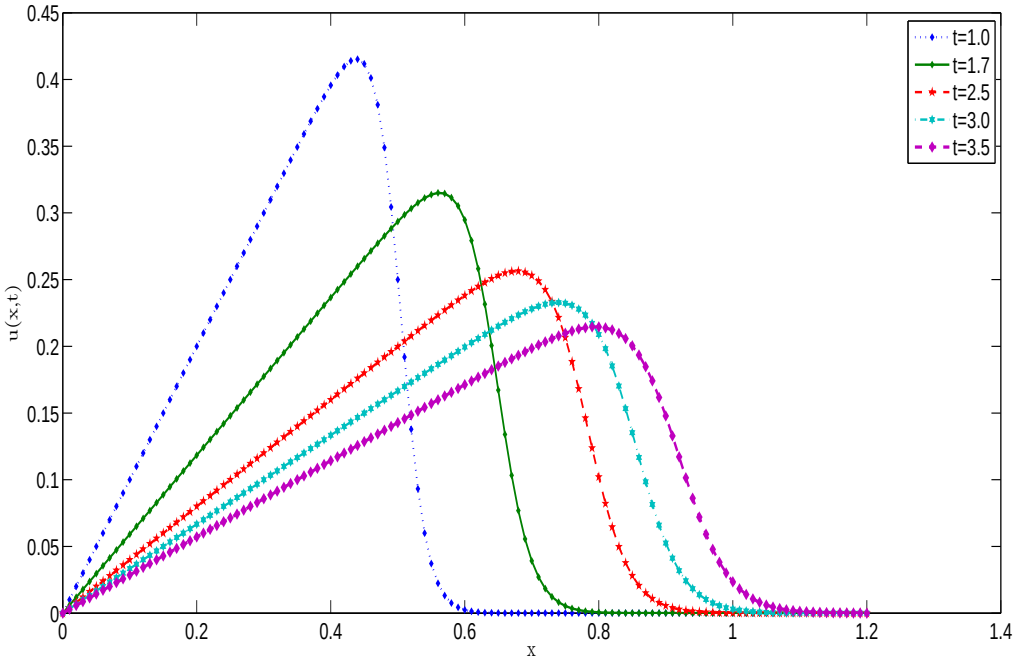


Figure 2.4: Physical behavior mECDQ solutions of Example 2.5.1 for $\nu = 0.05$ at different time levels with $h = \triangle t = 0.01$

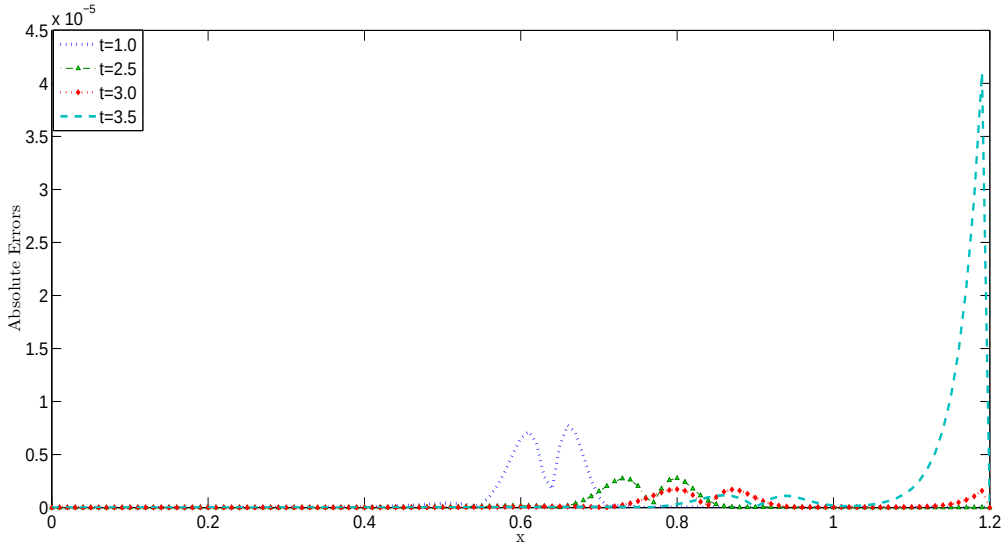


Figure 2.5: Absolute errors in mECDQ solutions of Example 2.5.1 for $\nu = 0.05$ at different time levels with $h = \triangle t = 0.01$

Table 2.3: Comparison of the $L_2$ and $L_\infty$ errors in mECDQ solutions of Example 2.5.1 for $\nu = 0.005$ at $t \leq 3.50$ with the errors obtained in earlier schemes

| | | | $t = 1.7$ | | $t = 2.4$ | | $t = 3.1$ | |
|---|---|---|---|---|---|---|---|---|
| Methods | $N_x$ | $\triangle t$ | $10^3 L_2$ | $10^3 L_\infty$ | $10^3 L_2$ | $10^3 L_\infty$ | $10^3 L_2$ | $10^3 L_\infty$ |
| mECDQ | 121 | 0.01 | 0.00101 | 0.0048 | 0.00062 | 0.00199 | 0.00070 | 0.00354 |
| [13] | 121 | 0.01 | 0.00191 | 0.0077 | 0.00086 | 0.00308 | 0.00065 | 0.00331 |
| PDQ [123] | 200 | 0.001 | 0.015 | 0.056 | 0.011 | 0.064 | 0.584 | 4.301 |
| CBCDQ [124] | 101 | 0.001 | | | 0.210 | 0.680 | 0.190 | 0.530 |
| QRTDQ [133] | 101 | 0.001 | 0.109 | 0.434 | 0.100 | 0.339 | 0.091 | 0.266 |
| BSFEM [49] | 50 | 0.1 | 0.857 | 2.576 | 0.423 | 1.242 | 0.230 | 0.680 |
| CSC [210] | 50 | 0.01 | 0.857 | 2.576 | 0.423 | 1.242 | 0.235 | 0.688 |
| QBCM1 [154] | 200 | 0.01 | 0.017 | 0.061 | 0.012 | 0.058 | 0.601 | 4.434 |
| QBCM2 [154] | 200 | 0.001 | 0.358 | 1.211 | 0.251 | 0.807 | 0.630 | 4.790 |
| Galerkin [282] | 200 | 0.01 | 0.857 | 2.576 | 0.423 | 1.242 | 0.235 | 0.688 |
| | | | | | $t = 2.5$ | | | |
| QBCM [55] | 200 | 0.01 | 0.0721 | 0.3115 | 0.0510 | 0.18902 | | |
| CBCM [55] | 200 | 0.01 | 2.4664 | 27.577 | 2.1118 | 25.1517 | | |
| QRKM [55] | 200 | 0.01 | 0.026 | 0.091 | 0.031 | 0.115 | | |
| | | | | | | | $t = 3.5$ | |
| [171] | 241 | 0.01 | 0.02520 | 0.0994 | 0.0151 | 0.0549 | 0.0117 | 0.0486 |
| $\beta$=0.5 [227] | 12001 | 0.01 | 0.38421 | 1.3473 | 0.49135 | 1.55470 | 0.52586 | 1.52196 |
| $\beta$=1 [227] | 12001 | 0.01 | 3.08966 | 10.404 | 2.72048 | 8.29747 | 2.12110 | 5.94321 |
| [13] | 121 | 0.01 | 0.00191 | 0.0078 | 0.0079 | 0.00275 | 0.00618 | 0.04335 |
| mECDQ | 121 | 0.01 | 0.00101 | 0.0048 | 0.00060 | 0.00179 | 0.00616 | 0.04317 |

*with* $\alpha = \beta = 1, \xi = \eta = -2$ *as follows*

$$\begin{cases} \dfrac{\partial u}{\partial t} = \dfrac{\partial^2 u}{\partial x^2} + 2u\dfrac{\partial u}{\partial x} - \dfrac{\partial uv}{\partial x}, & x \in (-\pi, \pi),\ t > 0 \\ \dfrac{\partial u}{\partial t} = \dfrac{\partial^2 v}{\partial x^2} + 2v\dfrac{\partial v}{\partial x} - \dfrac{\partial uv}{\partial x}, & x \in (-\pi, \pi),\ t > 0 \end{cases}$$

*where the values of* $\phi, \psi, g_1, g_2, g_3, g_4$ *can be extracted from the following exact solution of the problem as given in [182]*

$$u(x,t) = v(x,t) = e^{-t}\sin(x),\ x \in (-\pi, \pi), t \geq 0.$$

*The computed mECDQ solutions, the order of convergence and CPU time are compared with the solutions computed via LBM [147] are reported in Table 2.4. In Table 2.5, the computed solutions and CPU time are compared with the solutions computed via PDQM [170] and modified cubic B-spline collocation method [172]. The findings from the above tables show that computed solutions are more accurate than the solutions obtained by PDQM [170],*

*LBM [147] and modified cubic B-spline collocation method [172], the CPU time in mECDQ method is slightly more as compared to modified cubic B-spline collocation method [172] and is less than LBM. The order of convergence of mECDQ method more than the order of convergence of LBM [147].*

Table 2.4: Comparison of $L_2, L_\infty$ errors, CPU time and ROC of in mECDQ solutions with LBM [147] of Example 2.5.2 with $\alpha = \beta = 1, \xi = \eta = -2$ and $\triangle t = 0.0001$ at $t = 1$

| | mECDQ ($\lambda = -0.005$) | | | | | LBM [147] | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $N$ | $L_2$ | $ROC$ | $L_\infty$ | $ROC$ | $CPU$ | $L_2$ | $ROC$ | $L_\infty$ | $ROC$ | $CPU$ |
| 10 | 7.24E-03 | | 3.42E-03 | | 0.002 | 3.30E-02 | | 1.15E-02 | | 0.031 |
| 20 | 8.27E-04 | 3.13 | 4.09E-04 | 3.06 | 0.008 | 8.18E-03 | 2.01 | 3.01E-03 | 1.94 | 0.062 |
| 40 | 6.14E-05 | 3.75 | 3.24E-05 | 3.66 | 0.028 | 2.01E-03 | 2.03 | 7.38E-04 | 2.03 | 0.109 |
| 80 | 1.36E-05 | 2.18 | 6.47E-06 | 2.32 | 0.115 | 4.64E-04 | 2.11 | 1.71E-04 | 2.11 | 0.202 |

**Example 2.5.3.** *The third test case deals* $(1+1)D$ *nonlinear Burgers' equation* (2.1.1) *with* $\alpha = 1$, $\Omega_1 = [0, 2]$ *as considered in [13]:*

$$u(x,t) = \frac{2\pi\nu\left(\sin(\pi x)\exp(-\pi^2\nu^2 t) + 4\sin(2\pi x)\exp(-4\pi^2\nu^2 t)\right)}{\left(4 + \cos(\pi x)\exp(-\pi^2\nu^2 t) + 2\cos(2\pi x)\exp(-4\pi^2\nu^2 t)\right)} \tag{2.5.3}$$

*The values of* $\psi(x), \varsigma(x,t)$*can be extracted from the exact solution* (2.5.3).

*The numerical computation of Example 2.5.3 is performed with the parameter values* $h = 0.1, \nu = 10^{-2}$ *for the optimal value of* $\lambda$ *in* $[-10^4, 10^4]$ *with respect to* $L_\infty$ *error, which is* $\lambda = -4.3 \times 10^3$. *At* $t = 1, L_2$ *and* $L_\infty$ *errors with parameter values* $h = 0.1, \triangle t = 0.01$ *and* $\lambda = -4.3 \times 10^3$ *are compared in Table 2.6 with the results of [13, 171] for different values of* $\nu$. *The findings confirm that the proposed results more accurate than the results in [13, 171]. The computed solution behavior of this test example is depicted in Figure 2.6*

Table 2.5: The comparison of $L_2$, $L_\infty$ errors and CPU time in mECDQ solutions of Example 2.5.2 with $\alpha, \beta = 1, \xi, \eta = -2$ at $t = 1$

| | mCEDQ$\lambda = -0.095$ | | | [170] | | [172] | | |
|---|---|---|---|---|---|---|---|---|
| $\triangle t$ | $L_2(u)$ | $L_\infty(u)$ | CPU | $L_\infty(u)$ | $L_\infty(v)$ | $L_\infty(u)$ | $L_\infty(v)$ | CPU |
| 0.01 | 7.80E-05 | 2.84E-05 | 0.024 | 1.85E-03 | 1.85E-03 | 1.34E-04 | 1.34E-04 | 0.015 |
| 0.005 | 3.06E-05 | 1.13E-05 | 0.059 | 9.22E-04 | 9.22E-04 | 1.34E-04 | 1.34E-04 | 0.031 |
| 0.002 | 2.37E-05 | 1.66E-05 | 0.113 | 4.60E-04 | 4.60E-04 | 1.34E-04 | 1.34E-04 | 0.046 |
| 0.001 | 3.07E-05 | 2.18E-05 | 0.303 | 1.84E-04 | 1.84E-04 | 1.34E-04 | 1.34E-04 | 0.093 |
| 0.0005 | 3.50E-05 | 2.43E-05 | 0.424 | 9.19E-05 | 9.19E-05 | 1.52E-05 | 1.52E-05 | 0.171 |

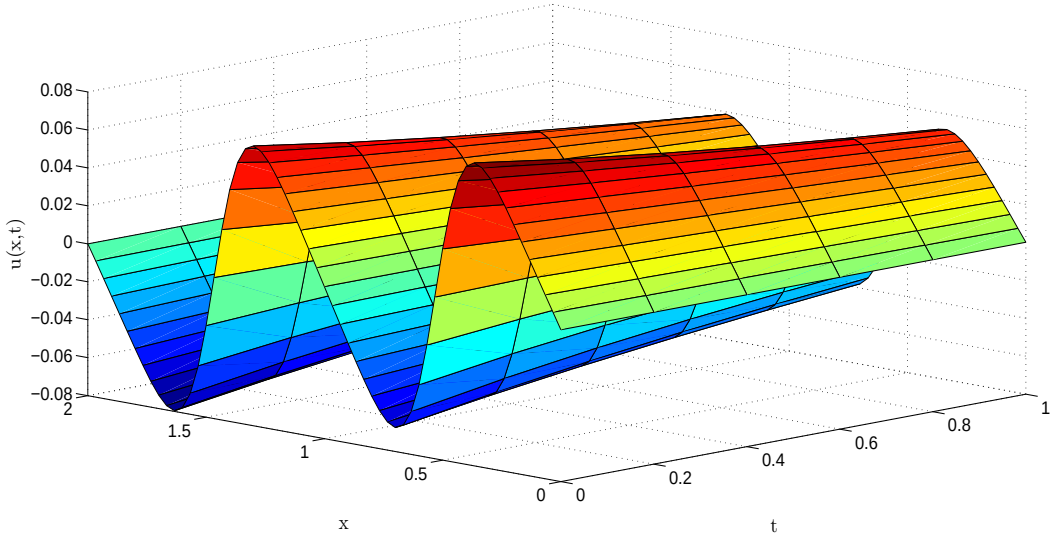


Figure 2.6: Behavior of mECDQ solutions of Example 2.5.3 for $\nu = 0.01$, $h = 0.02, \triangle t = 0.01$ at different time levels $t \leq 1$

Table 2.6: Comparison of obtained $L_2$ and $L_\infty$ errors with [13, 171].

| $\nu$ | Mittal and Jain [171] $h = 0.025$, $\triangle t = 10^{-3}$ | | MCB-DQM [13] $h = 0.1$, $\triangle t = 0.01$ | | mECDQ $h = 0.1$, $\triangle t = 0.01$ | |
|---|---|---|---|---|---|---|
| | $L_2$ | $L_\infty$ | $L_2$ | $L_\infty$ | $L_2$ | $L_\infty$ |
| $10^{-2}$ | 3.13E-02 | 2.66E-02 | 2.92E-02 | 2.63E-02 | 5.81E-04 | 7.99E-04 |
| $10^{-3}$ | 4.45E-04 | 3.59E-04 | 3.93E-04 | 3.45E-04 | 6.39E-06 | 8.55E-06 |
| $10^{-4}$ | 4.61E-06 | 3.72E-06 | 4.09E-06 | 3.55E-06 | 6.53E-08 | 8.61E-08 |
| $10^{-5}$ | 4.62E-08 | 3.74E-08 | 4.11E-08 | 3.56E-08 | 6.55E-10 | 8.61E-10 |
| $10^{-6}$ | 4.62E-10 | 3.74E-10 | 4.11E-10 | 3.56E-10 | 6.55E-12 | 8.61E-12 |

*at different time levels.*

### 2.5.2 $(2+1)$D nonlinear coupled viscous Burgers' equation

**Example 2.5.4.** *This case deals $(2+1)D$ nonlinear coupled viscous Burgers' equation 2.1.3 in $\Omega = [0,1]^2$ [11, 30, 275] with the following initial and boundary conditions*

$$\psi_1(x,y) = \sin(\pi x)\sin(\pi y),$$

$$\psi_2(x,y) = [\sin(\pi x) + \sin(2\pi x)]\ [\sin(\pi y) + \sin(2\pi y)],\ (x,y) \in \Omega,$$

$$\xi(x,y,t) = 0, \zeta(x,y,t) = 0,\ (x,y) \in \partial\Omega,\ t \geq 0,$$

*The numerical computation of Example 2.5.4 is performed for $\nu = 0.01, 1$, $\triangle t = 0.005, \lambda =$*

0.5 *at* $t \leq 1$ *for different grid points. The computed results are compared with the results of [11,30,275] in Tables 2.7 and 2.8. The findings from Table 2.7 and Table 2.8 show that the proposed results are agreed well with the results obtained by fourth-order compact schemes: ETDRK4-P13 [30]. The physical behavior at different time levels* $t = 0.04, 0.25, 0.5, 1$ *is depicted in Figure 2.7 for* $\nu = 0.01$, $N_x = N_y = 81$ *and* $\triangle t = 0.005$.

**Example 2.5.5.** *This test case deals with* $(2+1)D$ *nonlinear coupled viscous Burgers'*

Table 2.7: Comparison of mECDQ results of Example 2.5.4 at $t = 0.01$ for $Re = 1.0$

| | Points | [30] | | Method of line [11] | | FEM [11] | | mECDQ |
|---|---|---|---|---|---|---|---|---|
| | $N_x$ | 20 | 40 | 20 | 40 | 20 | 40 | 40 |
| | $\triangle t$ | 0.005 | 0.0050 | 0.005 | 0.00025 | 0.00067 | 0.0005 | 0.005 |
| u | (0.1, 0.1) | 0.07258 | 0.07253 | 0.07257 | 0.07253 | 0.07257 | 0.07252 | 0.072517 |
| | (0.2, 0.8) | 0.28846 | 0.28836 | 0.28846 | 0.28836 | 0.28842 | 0.28835 | 0.277556 |
| | (0.4, 0.4) | 0.72206 | 0.72178 | 0.72205 | 0.72178 | 0.72210 | 0.72179 | 0.721622 |
| | (0.7, 0.1) | 0.20113 | 0.20106 | 0.20112 | 0.20106 | 0.20113 | 0.20107 | 0.204770 |
| | (0.9, 0.9) | 0.07949 | 0.07947 | 0.07948 | 0.07947 | 0.07947 | 0.07946 | 0.079461 |
| v | (0.1, 0.1) | 0.43302 | 0.43162 | 0.43302 | 0.43173 | 0.44336 | 0.43178 | 0.431192 |
| | (0.2, 0.8) | -0.12386 | -0.12182 | -0.12387 | -0.12184 | -0.12366 | -0.12180 | -0.124340 |
| | (0.4, 0.4) | 1.65573 | 1.65336 | 1.65571 | 1.65335 | 1.65499 | 1.65316 | 1.652340 |
| | (0.7, 0.1) | 0.06571 | 0.06681 | 0.06571 | 0.06679 | 0.06621 | 0.06692 | 0.066822 |
| | (0.9, 0.9) | 0.01372 | 0.01349 | 0.01372 | 0.01349 | 0.01367 | 0.01349 | 0.013431 |

Table 2.8: Comparison of mECDQ results of Example 2.5.4 with the results in [30] and [275] for $Re = \frac{1}{\nu} = 100$

| Points | | mECDQ | | [30] | | [275] | |
|---|---|---|---|---|---|---|---|
| | | $N_x = 80, \triangle t = 0.005$ | | $N_x = 80, \triangle t = 0.002$ | | $N_x = 80, \triangle t = 0.002$ | |
| | | $t = 0.5$ | t=1.0 | $t = 0.5$ | t=1.0 | $t = 0.5$ | t=1.0 |
| (0.1, 0.1) | $u$ | 0.015095 | 0.007264 | 0.01510 | 0.00727 | 0.01509 | 0.00726 |
| | $v$ | 0.121633 | 0.055424 | 0.12169 | 0.05546 | 0.12162 | 0.05542 |
| (0.2,0.8) | $u$ | 0.158422 | 0.080754 | 0.15863 | 0.08073 | 0.15839 | 0.08076 |
| | $v$ | 0.987385 | 0.581761 | 0.99333 | 0.58211 | 0.98654 | 0.58111 |
| (0.4, 0.4) | $u$ | 0.128221 | 0.070449 | 0.12819 | 0.07043 | 0.12822 | 0.07045 |
| | $v$ | 0.700213 | 0.369000 | 0.70023 | 0.36899 | 0.70021 | 0.36900 |
| (0.7, 0.1) | $u$ | 0.133546 | 0.068174 | 0.13362 | 0.06823 | 0.13353 | 0.06816 |
| | $v$ | 0.099979 | 0.074458 | 0.09995 | 0.07444 | 0.09999 | 0.07446 |
| (0.8, 0.8) | $u$ | 0.563817 | 0.295711 | 0.56408 | 0.29573 | 0.56378 | 0.29571 |
| | $v$ | 1.185270 | 0.696786 | 1.18625 | 0.69686 | 1.18512 | 0.69677 |
| (0.9, 0.9) | $u$ | 0.281002 | 0.366676 | 0.27905 | 0.36783 | 0.28128 | 0.36648 |
| | $v$ | 0.221346 | 0.752660 | 0.21695 | 0.7544 | 0.22196 | 0.75237 |

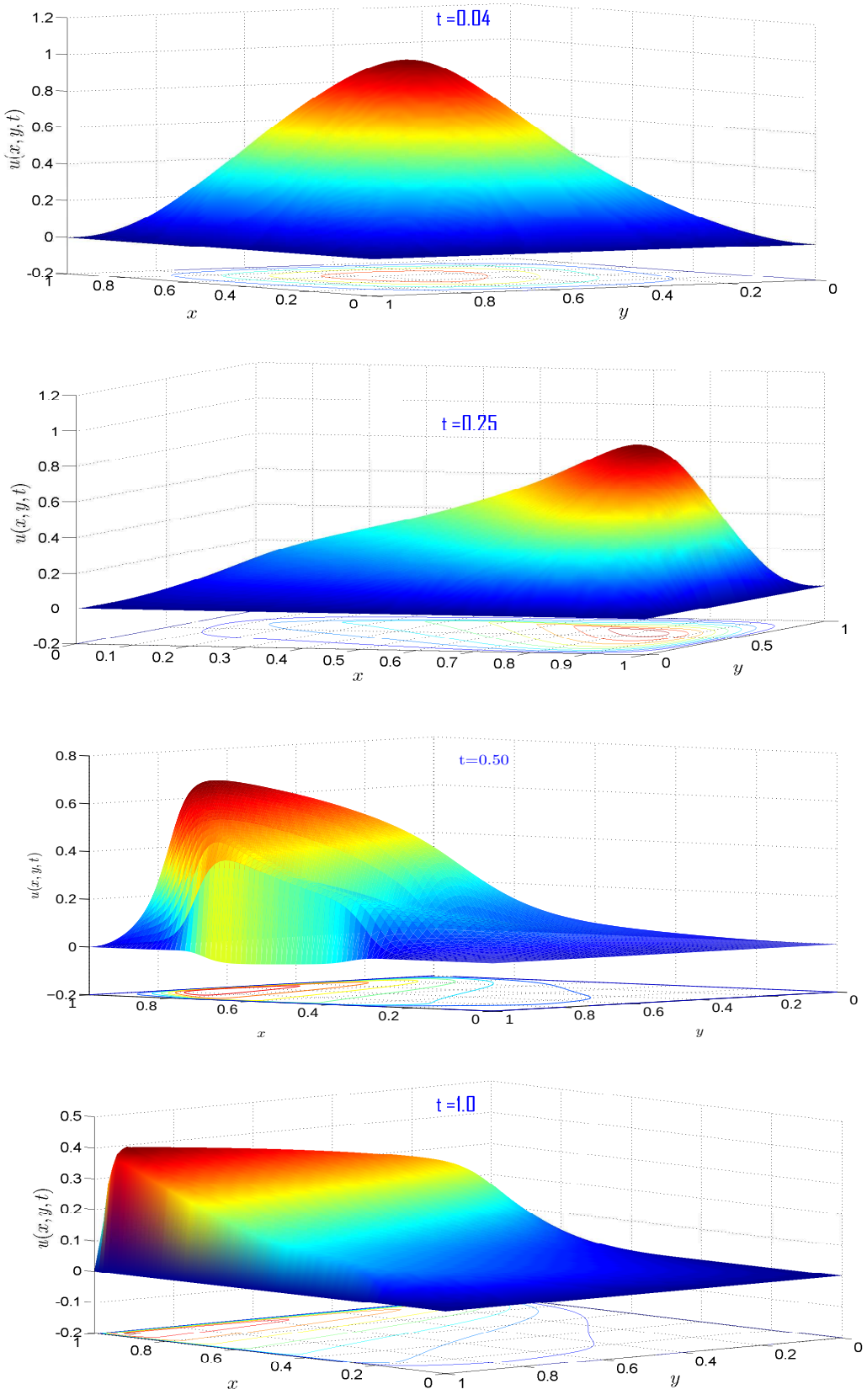


Figure 2.7: The behavior of Example 2.5.4 with $\nu = 0.01$, $N_x = N_y = 81$, $\triangle t = 0.005$ at different time levels $t \leq 1$

Table 2.9: Rate of convergence ($ROC$) for $u$ for different grid points $(N_x, N_x)$ with $\lambda = 0.52, \triangle t = 0.0005$ at $t = 1$

| $\lambda = 0.52$ | | | | | $\lambda = 0.4$ | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $N_x$ | $L_2$ | ROC | $L_\infty$ | ROC | $N_x$ | $L_2$ | ROC | $L_\infty$ | ROC |
| 5 | 1.5779E-02 | | 2.4630E-03 | | 4 | 1.0799E-02 | | 2.1600E-03 | |
| 10 | 1.2799E-03 | 3.62 | 1.2191E-04 | 4.34 | 8 | 2.6363E-03 | 2.03 | 2.8422E-04 | 2.93 |
| 20 | 2.0471E-04 | 2.64 | 1.2110E-05 | 3.33 | 16 | 3.5676E-04 | 2.89 | 2.6252E-05 | 3.44 |
| 40 | 3.4962E-05 | 2.55 | 1.1437E-06 | 3.40 | 32 | 6.2368E-05 | 2.52 | 2.4576E-06 | 3.42 |
| 80 | 5.5606E-06 | 2.65 | 1.0017E-07 | 3.51 | 64 | 1.0004E-05 | 2.64 | 2.1893E-07 | 3.49 |

Table 2.10: Rate of convergence ($ROC$) for $v$ for different grid points $(N_x, N_x)$ with $\lambda = 0.52, \triangle t = 0.0005$ at $t = 1$

| $\lambda = 0.52$ | | | | | $\lambda = 0.4$ | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $N_x$ | $L_2$ | ROC | $L_\infty$ | ROC | $N_x$ | $L_2$ | ROC | $L_\infty$ | ROC |
| 5 | 1.5779E-02 | | 2.4630E-03 | | 4 | 1.0799E-02 | | 2.1600E-03 | |
| 10 | 1.2799E-03 | 3.62 | 1.2191E-04 | 4.34 | 8 | 2.6363E-03 | 2.03 | 2.8422E-04 | 2.93 |
| 20 | 2.0471E-04 | 2.64 | 1.2110E-05 | 3.33 | 16 | 3.5676E-04 | 2.89 | 2.6252E-05 | 3.44 |
| 40 | 3.4962E-05 | 2.55 | 1.1437E-06 | 3.40 | 32 | 6.2368E-05 | 2.52 | 2.4576E-06 | 3.42 |
| 80 | 5.5606E-06 | 2.65 | 1.0017E-07 | 3.51 | 64 | 1.0004E-05 | 2.64 | 2.1893E-07 | 3.49 |

*equation* (2.1.3) *with the following exact solution as given in [65]*

$$u(x,y,t) = \frac{3}{4} - \frac{1}{4(1+\exp((-4x+4y-t))Re/32)}$$
$$v(x,y,t) = \frac{3}{4} + \frac{1}{4(1+\exp((-4x+4y-t))Re/32)}$$

*where* $\psi_1(x,y)$*and* $\psi_2(x,y)$ *on* $\Omega$*; and* $\zeta(x,y,y), \varsigma(x,y,t)$ *on* $\partial\Omega$ *can be computed from the exact solution for the computational domain* $\Omega = [0,1]^2$*.*

*The numerical computation of Example 2.5.5 is performed for* $\nu = 10^{-2}, \triangle t = 0.0005$ *and* $\lambda = 0.52$*. The computed* $L_2$ *and* $L_\infty$ *error norms for* $u$ *and* $v$ *are reported in Tables 2.9 and 2.10. It is found that the rate of convergence (ROC) for this example is cubic. The mCEDQ solution behaviors of* $u$ *and* $v$ *with* $\nu = 10^{-2}$ *and* $\lambda = 0.52$ *at time* $t = 0.5$ *are depicted in Figure 2.8 whereas the exact solution behaviors of* $u$ *and* $v$ *with the same parameters are depicted in Figure 2.9 at* $t = 0.5$*.*

**Example 2.5.6.** *This test case deals with* $(2+1)D$ *nonlinear coupled viscus Burgers' equa-*

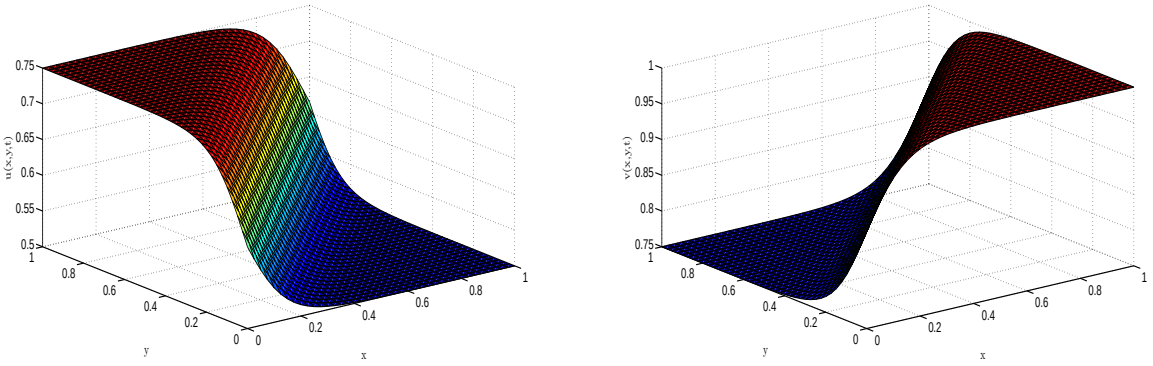

Figure 2.8: mECDQ solutions of Example 2.5.5 at $t = 0.05$ with $N_x = N_y = 41, \triangle t = 0.005$ and $\nu = 10^{-2}$.

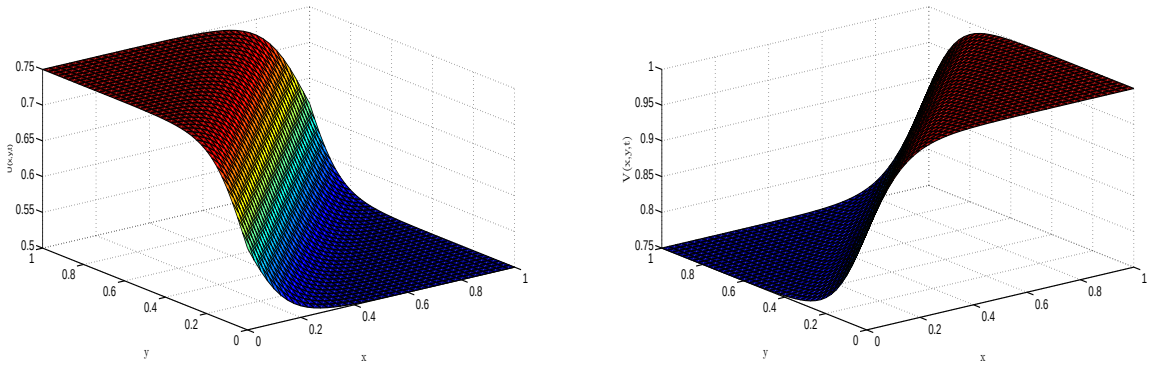

Figure 2.9: The exact solutions of Example 2.5.5 at $t = 0.05$ with $N_x = N_y = 41, \triangle t = 0.005$ and $\nu = 10^{-2}$.

*tion* (2.1.3) *with the following exact solutions*

$$u(x, y, t) = -e^{-2\nu t} \sin(x + y)$$
$$v(x, y, t) = e^{-2\nu t} \sin(x + y)$$

*where* $\psi_1(x, y)$ *and* $\psi_2(x, y)$ *on* $\Omega$*; and* $\zeta(x, y, t), \varsigma(x, y, t)$ *on* $\partial\Omega$ *can be computed from the exact solution for the computational domain* $\Omega = [-\pi, \pi]^2$.

*The numerical computation of Example 2.5.6 is performed for* $\triangle t = 0.001$ *and* $\nu = 2$ *at different grid points. The* $L_2, L_\infty$ *error norms are computed with the parameters:*$\lambda = 0.025$ *at* $t = 3.0$ *for different grid points, are reported in Table 2.11. Table 2.11 shows that for this problem mECDQ method converges linearly. The mCEDQ solution behavior of* $u$ *and* $v$ *with* $\nu = 0.5, N_x = N_y = 41$ *and* $\lambda = 0.25$ *at* $t = 3$ *is depicted in Figure 2.10 whereas the absolute errors are depicted in Figure 2.11.*

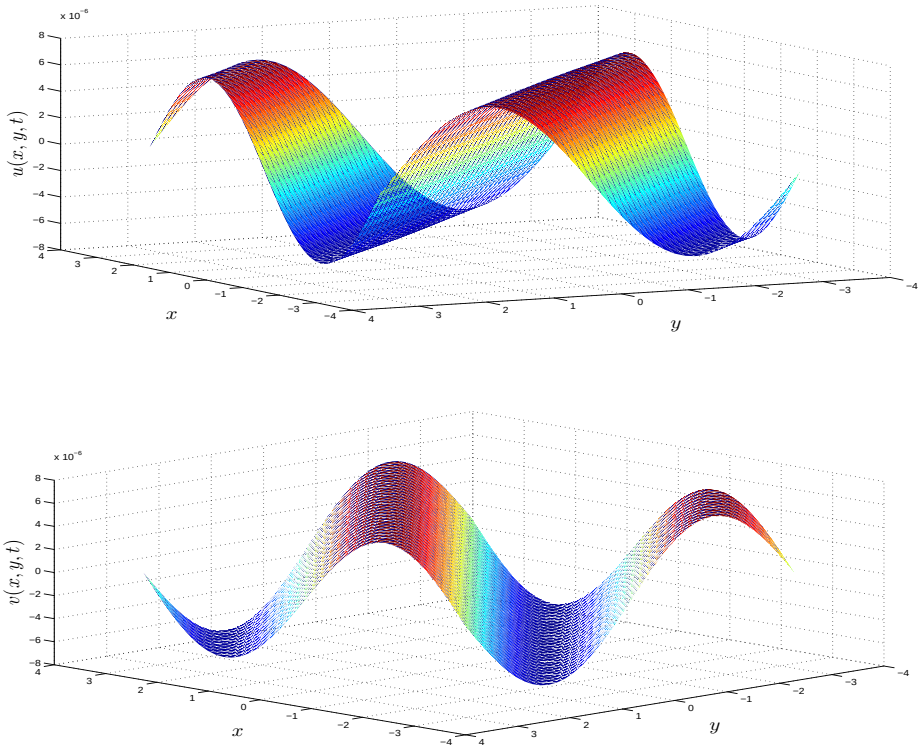


Figure 2.10: mECDQ solution of Example 2.5.6 at $t = 3.0$ with $N_x = N_y = 41$, $\triangle t = 0.001$ and $\nu = 0.05$.

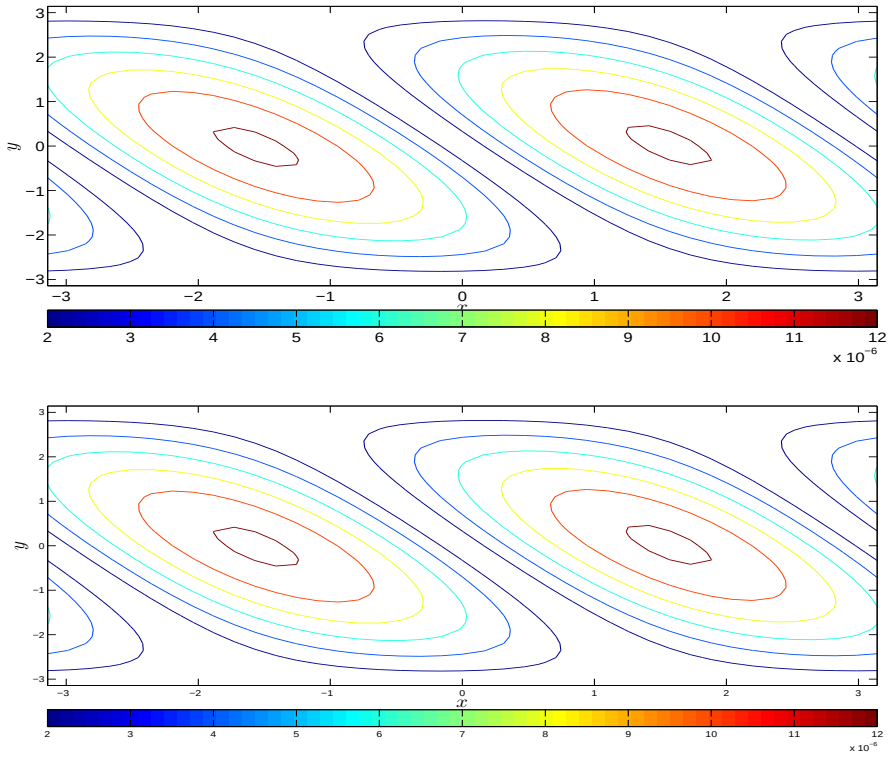


Figure 2.11: Contour plots of absolute errors in mECDQ solution of $u(upper), v(lower)$ of Example 2.5.6 at $t = 3.0$ with $N_x = N_y = 41$, $\triangle t = 0.001$ and $t = 0.5$.

Table 2.11: $L_\infty$, $L_2$ errors and ROC of Example 2.5.6 in $\Omega = [-\pi, \pi]^2$ for $\lambda = 0.25, \triangle t = 0.001$

| $N_x$ | u | | | v | | |
|---|---|---|---|---|---|---|
| | $L_2$ | $L_\infty$ | ROC ($L_\infty$) | $L_2$ | $L_\infty$ | ROC ($L_\infty$) |
| 10 | 7.20E-05 | 1.16E-06 | | 7.20E-05 | 1.16E-06 | |
| 20 | 4.29E-05 | 4.05E-07 | 0.72 | 4.29E-05 | 4.05E-07 | 1.52 |
| 40 | 3.90E-05 | 1.92E-07 | 1.08 | 3.90E-05 | 1.92E-07 | 1.08 |
| 60 | 3.84E-05 | 1.27E-07 | 1.01 | 3.84E-05 | 1.27E-07 | 1.01 |
| 80 | 3.82E-05 | 9.54E-08 | 1.00 | 3.82E-05 | 9.54E-08 | 1.00 |

## 2.6 Conclusions

This chapter deals with a new method "modified extended cubic B-spline differential quadrature method" developed for numerical computation of nonlinear partial differential equations. From the definition of the DQM, the modified extended cubic B-splines have been taken as a set of base functions to determine the weighting coefficients of the first-order derivative approximations. The polynomial based DQM has been used to determine the weighting coefficients of the second-order derivative approximations. The proposed mECDQ method is implemented for Burgers' equation in both one and two dimensions.

Section 2.3.1 confirms that the computational cost of both MCB-DQM [13] and mECDQ method with an optimal value of the free parameter is same. In Section 2.5, the accuracy and efficiency of mECDQ method have been measured by calculating $L_2$, $L_\infty$ error norms and the rate of convergence, which shows that mECDQ method generates very accurate solutions for Burgers' equation in both (1+1) and (2+1) dimensions. For instance, mECDQ method with suitable a value of $\lambda$ produces better results for $(1 + 1)$ dimensional Burgers' equation in comparison to MCB-DQM [13]. Moreover, for $(2+1)D$ coupled viscous Burgers' equation the mECDQ method produces results comparable to the recent result in [30]. The advantage of mECDQ method is its accuracy in comparison to MCB-DQM, low memory storage and easiness of the implementation. The mECDQ solutions of Burgers' equation are computed without transforming the equation and without any linearization technique. The proposed method is shown stable for Burgers' equation by performing the matrix stability analysis method for various grid values.

# Chapter 3

# Modified trigonometric cubic B-spline DQM for solving Burgers' equation

## 3.1 Introduction

This chapter deals with a new approach called "modified trigonometric cubic B-spline differential quadrature method (MTB-DQM)" in space together with a time integration algorithm to predict the approximate behavior of the initial value systems of nonlinear coupled viscous Burgers' equations with Dirichlet boundary conditions as given in (2.1.2) and (2.1.3) for one and two space dimension, respectively.

In the past years, a lot of efforts have been made for the accuracy and efficiency of various schemes for the numerical computation of Burgers' equation with different values of kinematic viscosity. The numerical computation of coupled viscous Burgers' equation in one space dimension has been done via various vigorous techniques, some of them are lattice Boltzmann method [147], differential quadrature method [170], fully implicit finite difference method [259], composite finite difference and Haar wavelets schemes [142], discrete adomian decomposition method [98], variational iteration method [31], cubic spline techniques [102], compact finite difference scheme of higher order [30, 215],Quartic B-spline collocation method [14] and many more. In 2016, Arora and Joshi [12] developed modified trigonometric cubic B-spline collocation method for the solution of one-dimensional

parabolic partial differential equation in biological processes. For more schemes for (2+1)D Buregers' equation, the interested readers are referred to [11, 49, 154, 210, 275, 282] and the references in chapter 2.

## 3.2 Description of MTB-DQM

This section deals with the description of modified trigonometric B-spline DQM for Burgers' equation. The computation domain $\Omega_1$ and $\Omega$ for one and two space dimension Burgers'equations are portioned uniformly in each direction as defined in chapter 2.

For the generic grids $x_i \in \Omega_1, (x_i, y_i) \in \Omega$, define

$$u_i := u_i(t) = u(x_i, t);\ u_{i,j} := u_{i,j}(t) = u(x_i, y_j, t)$$

Setting

$$a_1 = \frac{\sin^2\left(\frac{h_x}{2}\right)}{\sin(h_x)\sin\left(\frac{3h_x}{2}\right)};\quad a_2 = \frac{2}{1+2\cos(h_x)};\quad a_4 = \frac{3}{4\sin\left(\frac{3h_x}{2}\right)} = -a_3;$$
$$a_5 = \frac{3+9\cos(h_x)}{16\sin^2\left(\frac{h_x}{2}\right)\left(2\cos\left(\frac{h_x}{2}\right)+\cos\left(\frac{3h_x}{2}\right)\right)};\quad a_6 = \frac{3\cos^2(\frac{h_x}{2})}{\sin^2\left(\frac{h_x}{2}\right)(2+4\cos(h_x))}.$$

The trigonometric cubic B-spline function as defined in [12] is as follows

$$T_i = \frac{1}{\omega}\begin{cases} p^3(x_i), & x \in [x_i, x_{i+1}) \\ p(x_i)\{p(x_i)q(x_{i+2}) + p(x_{i+1})q(x_{i+3})\} + p^2(x_{i+1})q(x_{i+4}), & x \in [x_{i+1}, x_{i+2}) \\ q(x_{i+4})\{p(x_{i+1}q(x_{i+3}) + p(x_{i+2})q(x_{i+4})\} + p(x_i)q^2(x_{i+3}), & x \in [x_{i+2}, x_{i+3}) \\ q^3(x_{i+4}), & x \in [x_{i+3}, x_{i+4}) \end{cases} \tag{3.2.1}$$

where $p(x_m) = \sin(\frac{x-x_m}{2});\ q(x_m) = \sin(\frac{x_m-x}{2})$, $\omega = \sin\left(\frac{h_x}{2}\right)\sin(h_x)\sin\left(\frac{3h_x}{2}\right)$.

Noticed that the set $\{T_0, T_1, \ldots, T_N, T_{N+1}\}$ of trigonometric cubic B-splines form a base for the domain $\Omega_1$. The values of $T_{i,j} := T_i(x_j)$ and its derivatives $T'_{ij} = T'_i x_j; T''_{ij} = T''_i(x_j)$

at grid $x_j$ are respectively, read as:

$$T_{ij} = \begin{cases} a_2, & \text{if } i-j=0 \\ a_1, & \text{if } i-j=\pm 1 \\ 0, & \text{otherwise} \end{cases}; \tag{3.2.2}$$

$$T'_{ij} = \begin{cases} a_4, & \text{if } i-j=1 \\ a_3, & \text{if } i-j=-1 \\ 0, & \text{otherwise,} \end{cases};$$

$$T''_{ij} = \begin{cases} a_6, & \text{if } i-j=0 \\ a_5, & \text{if } i-j=\pm 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.2.3}$$

Analogous to [13,248], the set $\{\sigma_\ell : \ell \in \Delta_{N_x}\}$ of modified trigonometric cubic B-spline base functions over $\Omega_1$, are read as:

$$\begin{cases} \sigma_1(x) = T_1(x) + 2T_0(x) \\ \sigma_2(x) = T_2(x) - T_0(x) \\ \vdots \\ \sigma_j(x) = T_j(x), \text{ for } 2 < j < N_x - 1 \\ \vdots \\ \sigma_{N_x-1}(x) = T_{N_x-1}(x) - T_{N_x+1}(x) \\ \sigma_{N_x}(x) = T_{N_x}(x) + 2T_{N_x+1}(x) \end{cases} \tag{3.2.4}$$

### 3.2.1 Computation of the weighting coefficients

In order to evaluate the weighting coefficients $a_{ij}^{(1)}$ of for the first derivative (2.2.2), the modified trigonometric cubic B-spline $\sigma_m(x)$, $m \in \Delta_x$ are used. Setting $\sigma'_{mi} := \sigma'_m(x_i)$ and $\sigma_{mi} := \sigma_m(x_i)$. Accordingly, the approximation for the first-order derivative is given by:

$$\sigma'_{mi} = \sum_{\ell=1}^{N_x} a_{i\ell}^{(1)} \sigma_{m\ell}, \qquad m, i \in \Delta_{N_x}. \tag{3.2.5}$$

After setting $\Sigma = [\sigma_{m\ell}]$, $A = [a_{i\ell}^{(1)}]$, and $\Sigma' = [\sigma'_{mi}]$, then system (3.2.5) reduces to

$$\Sigma A^T = \Sigma'. \tag{3.2.6}$$

Eq.(3.2.2) and (3.2.4) yield the coefficient matrix $\Sigma$ of order $N_x$ as:

$$\Sigma = \begin{bmatrix} a_2+2a_1 & a_1 & & & & & & \\ 0 & a_2 & a_1 & & & & & \\ & a_1 & a_2 & a_1 & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & a_1 & a_2 & a_1 & & \\ & & & & a_1 & a_2 & 0 & \\ & & & & & a_1 & a_2+2a_1 \end{bmatrix}$$

and in particular the columns of the matrix $\Sigma'$ read:

$$\Sigma'[1] = \begin{bmatrix} 2a_4 \\ a_3-a_4 \\ 0 \\ \vdots \\ \\ 0 \\ 0 \end{bmatrix}, \Sigma'[2] = \begin{bmatrix} a_4 \\ 0 \\ a_3 \\ 0 \\ \vdots \\ \\ 0 \end{bmatrix}, \ldots, \Sigma'[N_x-1] = \begin{bmatrix} 0 \\ \vdots \\ \\ 0 \\ a_4 \\ 0 \\ a_3 \end{bmatrix}, \text{ and } \Sigma'[N_x] = \begin{bmatrix} 0 \\ \\ \vdots \\ \\ 0 \\ 2a_4 \\ a_3-a_4 \end{bmatrix}.$$

It is worth mentioning that the tridiagonal system (3.2.6) has a unique solution as $\Sigma$ is invertible. We use Thomas algorithm [151] to solve system (3.2.6) for weighting coefficients $a_{i\ell}^{(1)}, i, \ell \in \Delta_{N_x}$. Similarly, the coefficients $b_{i\ell}$ can be computed by considering the grid descretization in $y$ direction. Keeping computational cost in mind, we prefer Shu's $r$-th order (r$\geq$ 2) recursive formula (2.3.3) based on polynomial DQM [223, 239] to compute $a_{i\ell}^2, i, \ell \in \Delta_{N_x}$, $b_{i\ell}^{(2)}, i, \ell \in \Delta_{N_y}$.

### 3.2.2 Implementation of MTB-DQM for Burgers' equation

This section deals with the main goal of the chapter, the implementation of the proposed method for Burgers' equations in both one space and two space dimension.

#### A. $(1+1)$D nonlinear coupled viscous Burgers' equation

On putting the values of derivatives approximate from MTB-DQM, equation (2.1.2) can be re-written as:

$$\begin{cases} \dfrac{\partial u_i}{\partial t} = \displaystyle\sum_{j=1}^{N_x} a_{ij}^{(2)} u_j - \eta u_i \sum_{j=1}^{N_x} a_{ij}^{(1)} u_j - \alpha \left( u_i \sum_{j=1}^{N_x} a_{ij}^{(1)} v_j + v_i \sum_{j=1}^{N_x} a_{ij}^{(1)} u_j \right) \\ \dfrac{\partial v_i}{\partial t} = \displaystyle\sum_{j=1}^{N_x} a_{ij}^{(2)} v_j - \xi v_i \sum_{j=1}^{N_x} a_{ij}^{(1)} v_j - \beta \left( u_i \sum_{j=1}^{N_x} a_{ij}^{(1)} v_j + v_i \sum_{j=1}^{N_x} a_{ij}^{(1)} u_j \right) \\ u_i(0) = \phi(x_i),\ v_i(0) = \psi(x_i),\ i \in \Delta_{N_x} \end{cases} \tag{3.2.7}$$

On implementing the boundary conditions from (2.1.2), system (3.2.8) reduces to a set of first order ODEs as follows:

$$\begin{cases} \dfrac{\partial u_i}{\partial t} = \displaystyle\sum_{j=1}^{N_x-1} a_{ij}^{(2)} u_j - \eta u_i \sum_{j=1}^{N_x-1} a_{ij}^{(1)} u_j - \alpha \left( u_i \sum_{j=1}^{N_x-1} a_{ij}^{(1)} v_j + v_i \sum_{j=1}^{N_x-1} a_{ij}^{(1)} u_j \right) + F_i \\ \dfrac{\partial v_i}{\partial t} = \displaystyle\sum_{j=1}^{N_x-1} a_{ij}^{(2)} v_j - \xi v_i \sum_{j=1}^{N_x-1} a_{ij}^{(1)} v_j - \beta \left( u_i \sum_{j=1}^{N_x-1} a_{ij}^{(1)} v_j + v_i \sum_{j=1}^{N_x-1} a_{ij}^{(1)} u_j \right) + G_i \\ u_i(0) = \phi(x_i),\ v_i(0) = \psi(x_i),\ 1 < i \in < N_x \end{cases} \tag{3.2.8}$$

where $\eta_i = \eta u_i, \xi_i = \xi v_i, \alpha_i = \alpha u_i, \alpha_i' = \alpha v_i, \beta_i = \beta u_i, \beta_i' = \beta v_i$, and

$$F_i = (a_{i1}^{(2)} u_1 + a_{iN_x}^{(2)} u_{N_x}) - \eta_i (a_{i1}^{(1)} u_1 + a_{iN_x}^{(1)} u_{N_x}) - \alpha_i (a_{i1}^{(1)} v_1 + a_{iN_x}^{(1)} v_{N_x}) - \alpha_i' (a_{i1}^{(1)} u_1 + a_{iN_x}^{(1)} u_{N_x}),$$

$$G_i = (a_{i1}^{(2)} v_1 + a_{iN_x}^{(2)} v_{N_x}) - \xi_i (a_{i1}^{(1)} v_1 + a_{iN_x}^{(1)} v_{N_x}) - \beta_i (a_{i1}^{(1)} v_1 + a_{iN_x}^{(1)} v_{N_x}) - \beta_i' (a_{i1}^{(1)} u_1 + a_{iN_x}^{(1)} u_{N_x}).$$

**B.** $(2+1)$**D nonlinear coupled Burgers' equation**

Analogously, on implementing MTB-DQM to initial boundary value system of two space dimensional Burgers' equation (2.1.3) in space, we get

$$\begin{cases} \dfrac{\partial u_{ij}}{\partial t} = \nu \left( \displaystyle\sum_{k=2}^{N_x-1} a_{ik}^{(2)} u_{kj} + \sum_{k=2}^{N_y-1} b_{jk}^{(2)} u_{ik} \right) - u_{ij} \sum_{k=2}^{N_x-1} a_{ik}^{(1)} u_{kj} - v_{ij} \sum_{k=2}^{N_y-1} b_{jk}^{(1)} u_{ik} + F_{ij} \\ \dfrac{\partial v_{ij}}{\partial t} = \nu \left( \displaystyle\sum_{k=2}^{N_x-1} a_{ik}^{(2)} v_{kj} + \sum_{k=2}^{N_y-1} b_{jk}^{(2)} v_{ik} \right) - u_{ij} \sum_{k=2}^{N_x-1} a_{ik}^{(1)} v_{kj} - v_{ij} \sum_{k=2}^{N_y-1} b_{jk}^{(1)} v_{ik} + G_{ij} \\ u_{ij}(0) = \psi_1(x_i, x_j), v_{ij}(0) = \psi_2(x_i, x_j), 1 < i < N_x,\ 1 < j < N_y \end{cases} \tag{3.2.9}$$

where $u_{ij} = \tau_{ij}$ and $v_{ij} = \kappa_{ij}$

$$\begin{aligned} F_{ij} =& \nu \left( a_{i1}^{(2)} u_{1j} + a_{iN_x}^{(2)} u_{N_x j} + b_{j1}^{(2)} u_{i1} + b_{jN_y}^{(2)} u_{iN_y} \right) - \tau_{ij} \left( a_{i1}^{(1)} u_{1j} + a_{iN_x}^{(1)} u_{N_x j} \right) \\ & - \kappa_{ij} \left( b_{j1}^{(1)} u_{i1} + b_{jN_y}^{(1)} u_{iN_y} \right), \\ G_{ij} =& \nu \left( a_{i1}^{(2)} v_{1j} + a_{iN_x}^{(2)} v_{N_x j} + b_{j1}^{(2)} v_{i1} + b_{jN_y}^{(2)} v_{iN_y} \right) - \tau_{ij} \left( a_{i1}^{(1)} v_{1j} + a_{iN_x}^{(1)} v_{N_x j} \right) \\ & - \kappa_{ij} \left( b_{j1}^{(1)} v_{i1} + b_{jN_y}^{(1)} v_{iN_y} \right), \end{aligned} \tag{3.2.10}$$

The resulting initial value systems of first order ODEs (3.2.8) or (3.2.9) can be solved via many time integration algorithms. We prefer SSP-RK54 time integration algorithm (2.3.11) to solve these systems. The SSP-RK54 scheme allows low storage and large domain of absolute properties [133, 196, 197].

## 3.3 Stability analysis

The stability analysis for both dimension is being similar, we concerned with the stability analysis of $(2+1)$D coupled viscous Burgers' equation (3.2.9), only. Following [209], the term $u_{ij} = \tau_{ij}$, $v_{ij} = \kappa_{ij}$ in the nonlinear terms of equation (3.2.9) are assumed to be locally fixed. Then, Equation (3.2.9) can be re-written as

$$\frac{dU}{dt} = BU + H. \tag{3.3.1}$$

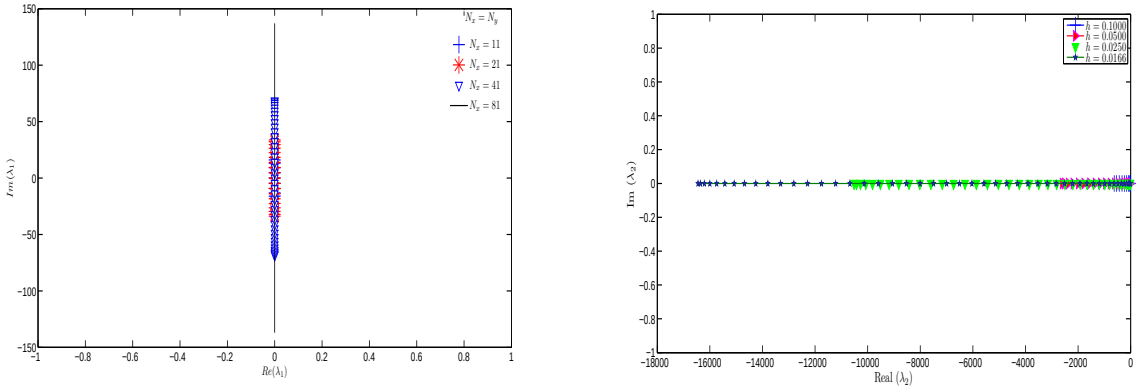


Figure 3.1: Eigenvalues $\lambda_1$ and $\lambda_2$ for different grid size $h$

where the definitions of $U, B, H$ and so, $A_r, B_r$ $(r = 1, 2)$ are same as defined in Section 2.4.

It is worth mentioning that stability of system (3.3.1) depends on the eigenvalues of the matrices B [226]. To claim the stability of the system (3.3.1), it is sufficient to claim that $\lambda_B \triangle t \in S$, (i.e., the stability region of SSP-RK54 algorithm, depicted in Figure 2.1) for each eigenvalue $\lambda_B$ of $B$. For more details, see [101, 135]. Similar to the previous chapter, the eigenvalues $\lambda_i$ of matrix $A_i$ $(i = 1, 2)$ computed for different step sizes $h_x = h_y = h$ are depicted in Figure 3.1, which concludes that each eigenvalue $\lambda_1$ of the matrices $A_1$ is pure imaginary while each eigenvalue $\lambda_2$ of the $A_2$ is real and negative.

Figure 3.1 and Equation (2.4.5) concludes that for given values of $h$ and $\nu, \exists \triangle t$ for which the value $\lambda_B \triangle t$ corresponding to each eigenvalue $\lambda_B = 2\nu\lambda_2 - \lambda_1(\tau_0 + \kappa_0)$ lies inside the stability region $S$ of SSP-RK54 algorithm. This shows that MTB-DQM produces stable solutions for two space dimensional coupled viscous Burgers' equation. Analogously, one can demonstrate that MTB-DQM method also produces stable solutions for one space dimensional coupled viscous Burger's equation.

## 3.4 Numerical results and discussion

The accuracy and consistency of the scheme is measured in terms of $L_2 := \sqrt{h \sum_{j=1}^{n} |U_j - U_j^*|^2}$ and $L_\infty := \max_{j=1}^{N_x} \left|u_j - u_j^*\right|$ error norms ($U_j \rightarrow$ exact solution and $U_j^* \rightarrow$ computed solution of at node $x_j$) for four test problems of $(1+1)D$ and $(2+1)D$ Burger's equation, which we computed by using by C++.

### 3.4.1 $(1+1)$D coupled Burgers' equation

**Example 3.4.1.** *The first test case deals with coupled viscous equation in one space dimension* (2.1.2) *with* $\alpha = \beta = 1, \xi = \eta = -2$ *as follows*

$$\begin{cases} \dfrac{\partial u}{\partial t} = \dfrac{\partial^2 u}{\partial x^2} + 2u\dfrac{\partial u}{\partial x} - \dfrac{\partial uv}{\partial x}, & x \in (-\pi, \pi),\ t > 0 \\ \dfrac{\partial u}{\partial t} = \dfrac{\partial^2 v}{\partial x^2} + 2v\dfrac{\partial v}{\partial x} - \dfrac{\partial uv}{\partial x}, & x \in (-\pi, \pi),\ t > 0 \end{cases}$$

*where the values of* $\phi, \psi, g_1, g_2, g_3, g_4$ *can be extracted from the following exact solution of the problem as given in [182]*

$$u(x,t) = v(x,t) = e^{-t}\sin(x),\ x \in (-\pi, \pi), t \geq 0.$$

The computed solutions, the order of convergence and CPU time are compared with the solutions computed via LBM [147], are reported in Table 3.1. In Table 3.2, the computed solutions and CPU time are compared with the solutions computed via PDQM [170] and a modified cubic B-spline collocation method [172]. The findings from the above tables show that computed solutions are more accurate than the solutions obtained by PDQM [170], LBM [147] and modified cubic B-spline collocation method [172], the CPU time in MTB-DQM is slightly more as compared to modified cubic B-spline collocation method [172] and is less than LBM. The order of convergence (ROC) of MTB-DQM is cubic in space while the order of convergence of LBM [147] is quadratic. It is evident from the Table 3.3 that the computed solutions are more accurate in comparison to FDM and LBM [147]. The comparison of the computed $L_2$ and $L_\infty$ errors in MTB-DQM solutions with the errors from existing schemes at different time levels $t \leq 3$, $\triangle t = 0.001$ and $N_x = 121$, is reported in Table 3.4. The findings show that the computed solutions are more accurate than the solutions obtained from the earlier methods in [142, 147, 168, 170, 202, 259], and have a good agreement with the exact solutions. $L_\infty$ and the absolute error norms at different time levels are depicted in Figure 3.2 whereas the solution behavior is depicted in Figure 3.3.

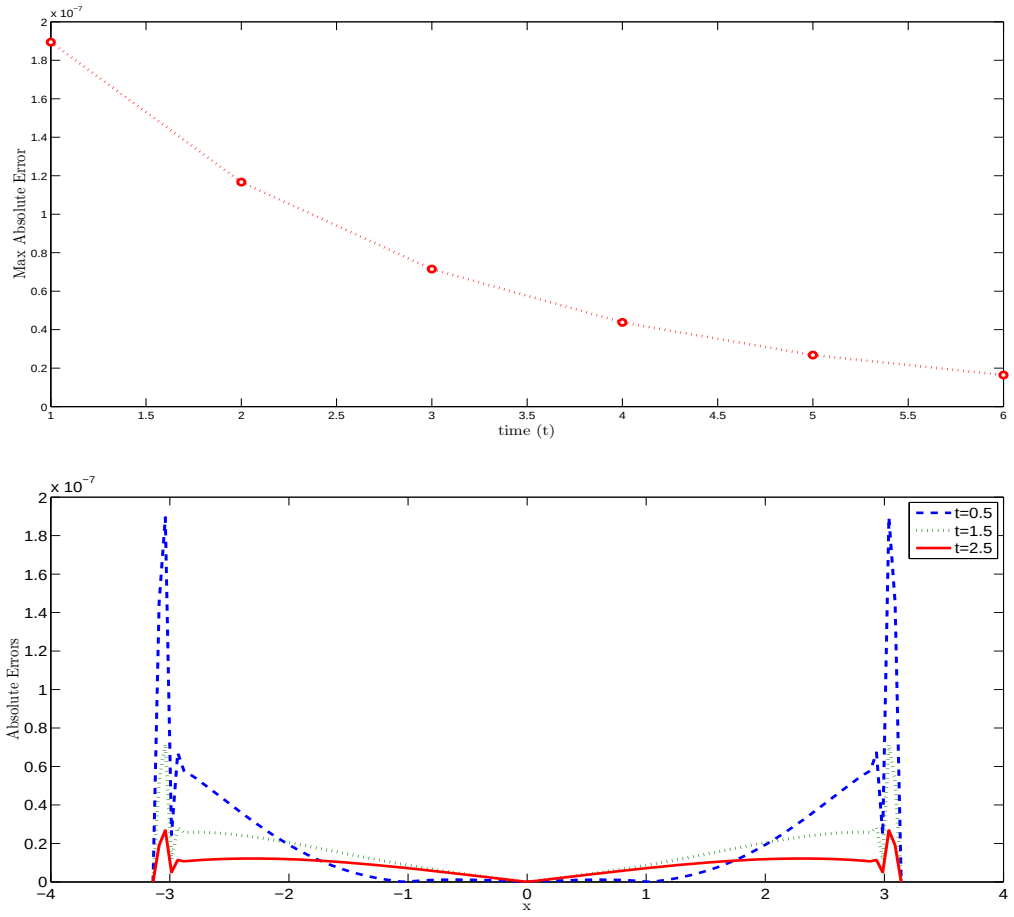


Figure 3.2: $L_\infty$ and absolute error norms in Example 3.4.1 for $u$ at different time levels

Table 3.1: Comparison of $L_2, L_\infty$ errors, CPU time and ROC of in MTB-DQM solutions with LBM [147] of Example 3.4.1 with $\alpha = \beta = 1, \xi = \eta = -2$ and $\triangle t = 0.0001$ at $t = 1$

| | MTB-DQM | | | | | LBM [147] | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $N$ | $L_2$ | ROC | $L_\infty$ | ROC | $CPU$ | $L_2$ | ROC | $L_\infty$ | ROC | $CPU$ |
| 10 | 6.69E-03 | | 3.27E-03 | | 0.004 | 3.30E-02 | | 1.15E-02 | | 0.031 |
| 20 | 8.77E-04 | 2.93 | 4.40E-04 | 2.89 | 0.010 | 8.18E-03 | 2.01 | 3.01E-03 | 1.94 | 0.062 |
| 40 | 1.03E-04 | 3.08 | 5.13E-05 | 3.10 | 0.040 | 2.01E-03 | 2.03 | 7.38E-04 | 2.03 | 0.109 |
| 80 | 8.01E-06 | 3.68 | 4.15E-06 | 3.63 | 0.136 | 4.64E-04 | 2.11 | 1.71E-04 | 2.11 | 0.202 |

Table 3.2: The comparison of $L_2$, $L_\infty$ errors and CPU time in MTB-DQM solutions in Example 3.4.1 with $\alpha, \beta = 1, \xi, \eta = -2$ at $t = 1$

| | MTB-DQM | | | [170] | | [172] | | |
|---|---|---|---|---|---|---|---|---|
| $\triangle t$ | $L_\infty(u)$ | $L_\infty(v)$ | CPU | $L_\infty(u)$ | $L_\infty(v)$ | $L_\infty(u)$ | $L_\infty(v)$ | CPU |
| 0.01 | 2.45E-05 | 2.45E-05 | 0.029 | 1.85E-03 | 1.85E-03 | 1.34E-04 | 1.34E-04 | 0.015 |
| 0.005 | 3.88E-06 | 3.88E-06 | 0.052 | 9.22E-04 | 9.22E-04 | 1.34E-04 | 1.34E-04 | 0.031 |
| 0.002 | 1.94E-05 | 1.94E-05 | 0.252 | 4.60E-04 | 4.60E-04 | 1.34E-04 | 1.34E-04 | 0.046 |
| 0.001 | 2.44E-05 | 2.44E-05 | 0.239 | 1.84E-04 | 1.84E-04 | 1.34E-04 | 1.34E-04 | 0.093 |
| 0.0005 | 2.69E-05 | 2.69E-05 | 0.423 | 9.19E-05 | 9.19E-05 | 1.52E-05 | 1.52E-05 | 0.171 |

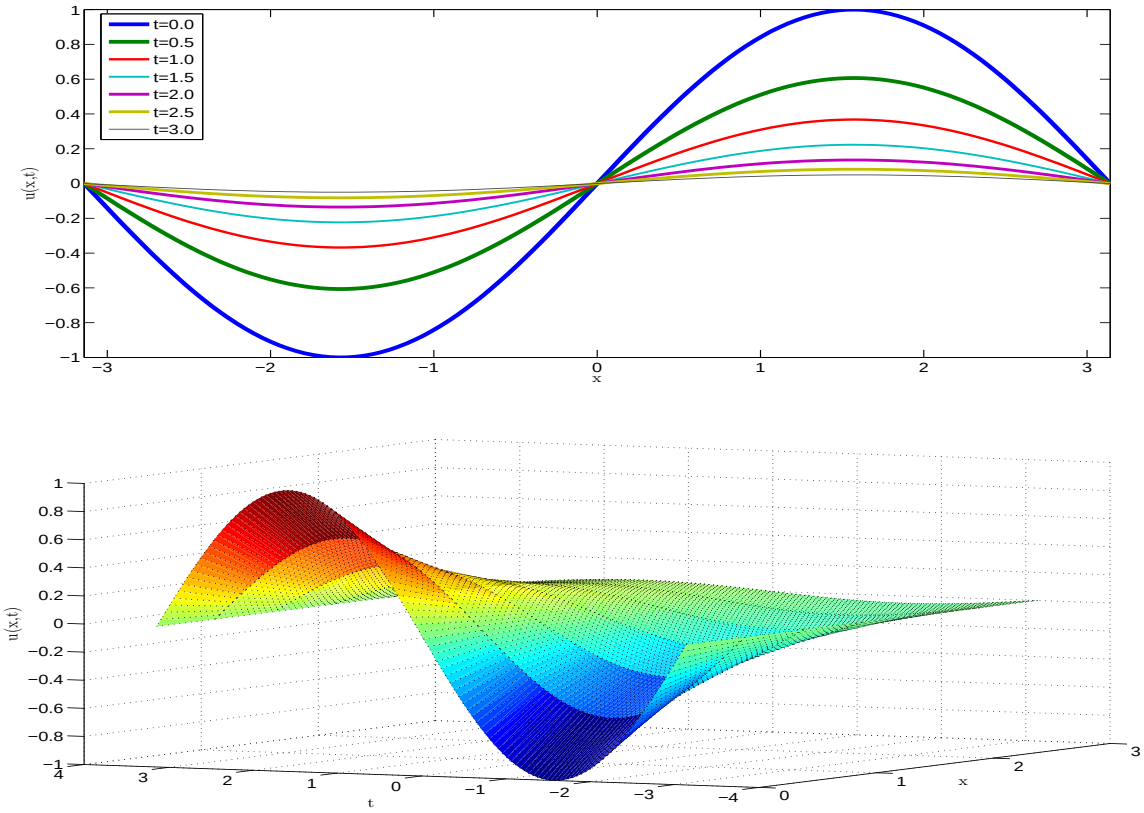


Figure 3.3: The 2D plots and surface behavior of MTB-DQM solution of Example 3.4.1 for $u$ at $t \leq 3.0$ for $N_x = 121; \triangle t = 0.001$

Table 3.3: Comparison of errors in MTB-DQM, FDM & LBM [147] for $\triangle t = 0.001$, $N = 64$

| | $L_2$ | | | $L_\infty$ | | |
|---|---|---|---|---|---|---|
| $t = 1$ | MTB-DQM | LBM [147] | FDM [147] | MTB-DQM | LBM [147] | FDM [147] |
| 0.1 | 8.44E-06 | 3.03E-05 | 8.03E-05 | 2.16E-05 | 2.75E-05 | 7.27E-05 |
| 0.5 | 1.45E-05 | 1.52E-04 | 4.02E-04 | 1.62E-05 | 9.20E-05 | 2.44E-04 |
| 1.0 | 2.02E-05 | 3.03E-04 | 8.03E-04 | 1.02E-05 | 1.12E-04 | 2.95E-04 |
| 2.0 | 3.30E-05 | 6.07E-04 | 1.61E-03 | 4.20E-06 | 8.21E-05 | 2.18E-04 |
| 5.0 | 7.61E-05 | 1.52E-03 | 4.02E-03 | 4.87E-07 | 1.02E-05 | 2.71E-05 |
| 10.0 | 1.50E-04 | 3.04E-03 | 8.06E-03 | 6.64E-09 | 1.38E-07 | 3.66E-07 |
| 20.0 | 3.00E-04 | 6.09E-03 | 1.62E-02 | 6.10E-13 | 1.25E-11 | 3.34E-11 |

### 3.4.2 $(2+1)$D coupled Burgers' equation

**Example 3.4.2.** *This test case deals with* $(2+1)D$ *Burgers' equation* (2.1.3) *in* $\Omega = [0.5, 0.5]^2$ *as in [98] with* $\psi_1(x,y) = x + y, \psi_2(x,y) = x - y$ *and*

$$\begin{cases} u(0,y,t) = \dfrac{y}{1-2t^2}, & u(0.5,y,t) = \dfrac{0.5+y-t}{1-2t^2} \\ v(0,y,t) = \dfrac{-y-2yt}{1-2t^2}, & v(0.5,y,t) = \dfrac{0.5-y-2yt}{1-2t^2}, \end{cases} \quad 0 \leq y \leq 0.5, t \geq 0,$$

Table 3.4: Comparison with earlier schemes for solution of $u$ in Example 3.4.1 with $\alpha = \beta = 1, \xi = \eta = -2$ at $t = 1$

| Schemes | $N_x$ | $\triangle t$ | $t = 0.5$ | | $t = 1$ | | $t = 2$ | $t = 3$ |
|---|---|---|---|---|---|---|---|---|
| | | | $L_2$ | $L_\infty$ | $L_2$ | $L_\infty$ | $L_2$ | $L_\infty$ |
| MTB-DQM | 121 | 0.001 | 2.00E-07 | 7.82E-08 | 1.21E-07 | 8.45E-08 | 4.42E-08 | 1.62E-08 |
| DQM[24] | | 0.01 | 1.52E-04 | | 1.84E-04 | | 1.35E-04 | 7.46E-05 |
| CBC[16] | 400 | 0.001 | 6.22E-06 | 1.02E-05 | 7.56E-06 | 2.04E-05 | | |
| CBC[16] | 200 | 0.001 | 4.10E-05 | 2.49E-05 | 8.21E-05 | 0.00003 | $t = 0.1$ | |
| FPM [14] | 128 | 0.0001 | | | 1.16E-05 | 2.88E-05 | $L_\infty$ | $L_2$ |
| FFID[25] | 200 | 0.001 | 1.79E-04 | 2.94E-04 | 2.17E-04 | 5.91E-04 | 5.30E-05 | 5.86E-05 |
| CFDH[26] | 16 | 0.001 | 7.27E-06 | 1.05E-08 | 2.38E-05 | 5.03E-07 | 3.77E-05 | 3.27E-08 |
| MTB-DQM | 121 | 0.001 | 2.00E-07 | 7.82E-08 | 1.21E-07 | 8.45E-08 | 3.01E-07 | 7.52E-08 |

$$\begin{cases} u(x,0,t) = \dfrac{x-2xt}{1-2t^2}, \quad u(x,0.5,t) = \dfrac{x-0.5-2xt}{1-2t^2} \\ v(x,0,t) = \dfrac{x}{1-2t^2}, \quad v(x,0.5,t) = \dfrac{x-0.5-2t}{1-2t^2}, \ 0 \le x \le 0.5, t \ge 0, \end{cases}$$

*The exact solutions given in [11] as follows:*

$$u(x,y,t) = \frac{x+y-2xt}{1-2t^2}; \quad v(x,y,t) = \frac{x-y-2yt}{1-2t^2}, \ (x,y) \in \Omega, t > 0.$$

The MTB-DQM solutions are computed for $h_x = h_y = 0.025$ and $\triangle t = 10^{-4}$ at time $t = 0.1$, and compared them (the values of $u$ and $v$) with the exact solutions for $Re = 1/\nu = 80$ in Tables 3.5 and 3.6 for different grid points and time levels $t = 0.1, 0.3, 0.5$ . $L_2, L_\infty$ errors for $u$ and $v$ are found same, reported in Table 3.7. The finding from the above tables show that the computed results are agreed well with the exact solutions. The physical behavior of MTB-DQM solutions of Example 3.4.2 for $u$ and $v$ components are depicted for $Re = 100$ in Figure 3.4 at time $t = 0.1$ whereas for $Re = 80$ in Figure 3.5 at time $t = 0.5$.

**Example 3.4.3.** *This test case deals with* $(2+1)D$ *initial value coupled viscous Burgers' equations* (2.1.3) *in* $\Omega = [0, 0.5]^2$ *with* $\psi_1(x,y) = \sin(\pi x) + \cos(\pi y)$, $\psi_2(x,y) = x + y$ *subject to the following boundary conditions*

$$\begin{cases} u(0,y,t) = \cos(\pi y), \quad u(0.5,y,t) = 1 + \cos(\pi y) \\ v(0,y,t) = y, \qquad\quad v(0.5,y,t) = 0.5 + y, \ 0 \le y \le 0.5, t \ge 0, \end{cases}$$

$$\begin{cases} u(x,0,t) = 1+\sin(\pi x), & u(x,0.5,t) = \sin(\pi x) \\ v(x,0,t) = x, & v(x,0.5,t) = x+0.5,\ 0 \le x \le 0.5, t \ge 0. \end{cases}$$

The numerical solutions are computed at $t = 0.1$ for uniform mesh grid $h_x = h_y = 0.025$ and $\triangle t = 10^{-4}$.

The MTB-DQM solutions of $u$ and $v$ components in Example 3.4.3 with $Re = 50$ are compared with the solutions in [102, 261, 268] and also with exact solutions taking grid size $20 \times 20$ and various time levels, see Table 3.8. The physical MTB-DQM solution behavior of $u$ and $v$ components in Example 3.4.3 with $\nu = 0.01$ is depicted in Figure 3.6 at different time levels $t = 1, 2, 3$.

**Example 3.4.4.** *The last test case deals with* $(2+1)D$ *coupled viscous Burgers' equation* (2.1.3) *in* $\Omega = [0,1]^2$ *with the following exact solution as in [65]*

$$\begin{aligned} u(x,y,t) &= \frac{3}{4} - \frac{1}{4\left[1+\exp\left((-4x+4y-t)\frac{Re}{32}\right)\right]} \\ v(x,y,t) &= \frac{3}{4} + \frac{1}{4\left[1+\exp\left((-4x+4y-t)\frac{Re}{32}\right)\right]} \end{aligned} \tag{3.4.1}$$

*where* $\psi_1, \psi_2$ *on* $\Omega$ *and* $\xi, \zeta$ *on* $\partial\Omega$ *can be computed from the exact solution* (3.4.1).

The numerical solutions is performed for $\nu, \triangle t = 0.0001$. The computed $L_2$ and $L_\infty$ errors are compared with the errors in mECDQ [248], Expo-MCB-DQM [268] solutions, in

Table 3.5: Comparison of MTB-DQM solutions with exact solutions of $u(x,y,t)$ for $Re = 80, \triangle t = 10^{-4}$ at different time levels

| Mesh | t=0.1 | | t=0.3 | | t=0.5 | |
|---|---|---|---|---|---|---|
| | Num. | Exact | Num. | Exact | Num. | Exact |
| 0.1,0.1 | 0.183673 | 0.183673 | 0.170732 | 0.170732 | 0.20004 | 0.20004 |
| 0.3,.01 | 0.346939 | 0.346939 | 0.268259 | 0.268259 | 0.19996 | 0.19996 |
| 0.2,0.2 | 0.367347 | 0.367347 | 0.341465 | 0.341465 | 0.40008 | 0.40008 |
| 0.4,0.2 | 0.530612 | 0.530612 | 0.438991 | 0.438991 | 0.40000 | 0.40000 |
| 0.5,0.3 | 0.714286 | 0.714286 | 0.609723 | 0.609723 | 0.60004 | 0.60004 |
| 0.1,0.3 | 0.387755 | 0.387755 | 0.414670 | 0.414670 | 0.60020 | 0.60020 |
| 0.3,0.4 | 0.653061 | 0.653061 | 0.634166 | 0.634166 | 0.80020 | 0.80020 |
| 0.2,0.4 | 0.571429 | 0.571429 | 0.585403 | 0.585403 | 0.80024 | 0.80024 |
| 0.4,0.5 | 0.836735 | 0.836735 | 0.804898 | 0.804898 | 1.00024 | 1.00024 |
| 0.5,0.5 | 0.918367 | 0.918367 | 0.853662 | 0.853662 | 1.00020 | 1.00020 |

Table 3.6: Comparison of the MTB-DQM solutions for $v(x, y, t)$ of Example 3.4.2 with $Re = 80, \triangle t = 10^{-4}$ with exact solutions at different time levels

| Mesh | t=0.1 | | t=0.3 | | t=0.5 | |
|---|---|---|---|---|---|---|
| | Num. | Exact | Num. | Exact | Num. | Exact |
| 0.1, 0.1 | -0.02041 | -0.02041 | -0.07321 | -0.07321 | -0.20012 | -0.20012 |
| 0.3, 0.1 | 0.183673 | 0.183673 | 0.170732 | 0.170732 | 0.20004 | 0.20004 |
| 0.2, 0.2 | -0.04082 | -0.04082 | -0.14641 | -0.14641 | -0.40024 | -0.40024 |
| 0.4, 0.2 | 0.163265 | 0.163265 | 0.097527 | 0.097527 | -0.00008 | -0.00008 |
| 0.5, 0.3 | 0.142857 | 0.142857 | 0.024321 | 0.024321 | -0.2002 | -0.2002 |
| 0.1, 0.3 | -0.23980 | -0.2398 | -0.46356 | -0.46356 | -1.00052 | -1.00052 |
| 0.3, 0.4 | -0.18367 | -0.18367 | -0.41479 | -0.41479 | -1.00056 | -1.00056 |
| 0.2, 0.4 | -0.28571 | -0.28571 | -0.53676 | -0.53676 | -1.20064 | -1.20064 |
| 0.4, 0.5 | -0.20408 | -0.20408 | -0.48800 | -0.48800 | -1.20068 | -1.20068 |
| 0.5, 0.5 | -0.10204 | -0.10204 | -0.36603 | -0.36603 | -1.0006 | -1.0006 |

Table 3.7: Comparison of the MTB-DQM solutions for $v(x, y, t)$ of Example 3.4.2 with $Re = 80, \triangle t = 10^{-4}$ with exact solutions at different time levels

| $(N_x, N_y)$ | $u$ | | | | $v$ | | | |
|---|---|---|---|---|---|---|---|---|
| | t=0.5 | | t=0.01 | | t=0.5 | | t=0.01 | |
| | $L_2$ | $L_\infty$ | $L_2$ | $L_\infty$ | $L_2$ | $L_\infty$ | $L_2$ | $L_\infty$ |
| (4, 4) | 6.23E-08 | 2.40E-08 | 1.48E-09 | 4.83E-10 | 6.23E-08 | 2.40E-08 | 1.48E-09 | 4.83E-10 |
| (8, 8) | 4.21E-09 | 1.21E-09 | 1.11E-10 | 2.27E-11 | 4.21E-09 | 1.21E-09 | 1.11E-10 | 2.27E-11 |
| (17, 17) | 2.66E-10 | 4.36E-11 | 7.55E-12 | 9.03E-13 | 2.66E-10 | 4.36E-11 | 7.55E-12 | 9.03E-13 |
| (32, 32) | 1.61E-11 | 1.40E-12 | 5.10E-13 | 3.13E-14 | 1.61E-11 | 1.40E-12 | 5.10E-13 | 3.13E-14 |
| (44, 44) | 4.13E-12 | 2.64E-13 | 1.60E-13 | 6.71E-15 | 4.13E-12 | 2.64E-13 | 1.60E-13 | 6.71E-15 |
| (64, 64) | 6.28E-13 | 2.28E-14 | 5.43E-14 | 1.80E-15 | 6.28E-13 | 2.28E-14 | 5.43E-14 | 1.80E-15 |

Table 3.8: Comparison of solutions of Example 3.4.3 with $Re = 50$ for $N_x = N_y = 20, \triangle t = 10^{-4}$ at $t = 0.625$

| Mesh | $u$ | | | | $v$ | | | |
|---|---|---|---|---|---|---|---|---|
| | MTB-DQM | FDM [261] | Expo-MCB -DQM [268] | [102] | MTB-DQM | FDM [261] | Expo-MCB -DQM [268] | [102] |
| 0.1,0.1 | 0.97056 | 0.97146 | 0.97056 | 0.97258 | 0.09842 | 0.09869 | 0.09842 | 0.09773 |
| 0.3,0.1 | 1.15152 | 1.15280 | 1.15152 | 1.16214 | 0.14107 | 0.14158 | 0.14107 | 0.14039 |
| 0.2,0.2 | 0.86244 | 0.86308 | 0.86243 | 0.86281 | 0.16732 | 0.16754 | 0.16732 | 0.16660 |
| 0.4,0.2 | 0.98078 | 0.97985 | 0.98078 | 0.96483 | 0.17223 | 0.17111 | 0.17223 | 0.17397 |
| 0.1,0.3 | 0.66336 | 0.66316 | 0.66335 | 0.66318 | 0.26380 | 0.26378 | 0.26380 | 0.26294 |
| 0.3,0.3 | 0.77226 | 0.77233 | 0.77226 | 0.77030 | 0.22653 | 0.22655 | 0.22653 | 0.22463 |
| 0.2,0.4 | 0.58273 | 0.58181 | 0.58273 | 0.58070 | 0.32935 | 0.32851 | 0.32935 | 0.32402 |
| 0.4,0.4 | 0.76179 | 0.75862 | 0.76179 | 0.74435 | 0.32884 | 0.32502 | 0.32884 | 0.31822 |

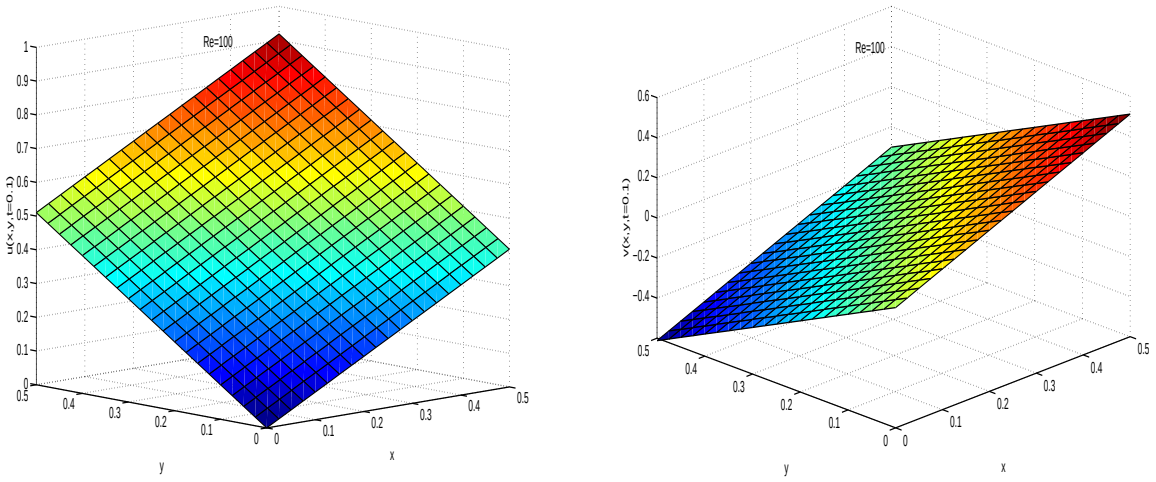


Figure 3.4: Behavior of MTB-DQM solutions of Example 3.4.2 with $Re = 100$ for $u, v$ at $t = 0.1$

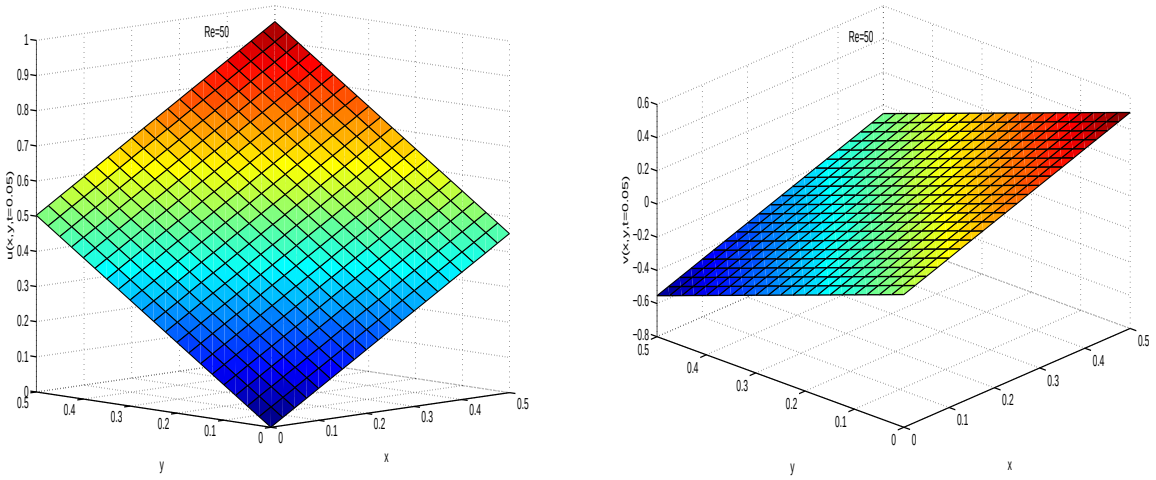


Figure 3.5: Behavior of MTB-DQM solutions of Example 3.4.2 with $Re = 80$ for $u, v$ at $t = 0.0.05$

Table 3.9. It is evident that the proposed MTB-DQM solutions comparable to [248, 268], and are in good agreement with the exact solutions. The MTB-DQM solutions and exact solutions of $u$ and $v$ at $t = 1$ in Problem 3.4.4 with $\nu = 10^{-2}$are depicted in Figure 3.7 and Figure 3.8, respectively.

Table 3.9: Comparison of the errors, ROC for $(u, v)$ of Example 3.4.4 with $Re = 100, t = 0.0001$ at time $t = 1.0$ for different grids $N_x = N_y$

| $N_x$ | mECDQ [248] $\lambda = 0.52$ | | Exp-MCB-DQM [268],$p = 10$ | | MTB-DQM | | mECDQ [248] $\lambda = 0.52$ | | Exp-MCB-DQM [268],$p = 10$ | | MTB-DQM | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $L_2$ | ROC | $L_2$ | ROC | $L_2$ | ROC | $L_2$ | ROC | $L_2$ | ROC | $L_2$ | ROC |
| Solutions for $u$ | | | | | | | | | | | | |
| 4 | 1.0799E-02 | | 1.5865E-02 | | 1.6451E-02 | | 2.1600E-03 | | 2.3325E-03 | | 2.8958E-03 | |
| 8 | 2.6363E-03 | 2.03 | 1.8037E-03 | 3.13 | 1.9330E-03 | 3.08 | 2.8422E-04 | 2.93 | 1.6816E-04 | 3.79 | 1.9644E-04 | 3.89 |
| 16 | 3.5676E-04 | 2.89 | 3.8329E-04 | 2.23 | 3.9504E-04 | 2.29 | 3.6252E-05 | 3.44 | 1.9610E-05 | 3.10 | 2.0508E-05 | 3.26 |
| 32 | 6.2368E-05 | 2.52 | 8.0461E-05 | 2.52 | 8.1200E-05 | 2.28 | 2.4576E-06 | 3.42 | 2.1967E-06 | 3.15 | 2.2208E-06 | 3.21 |
| 64 | 1.0004E-05 | 2.64 | 1.5355E-05 | 2.38 | 1.5323E-05 | 2.41 | 2.1893E-07 | 3.49 | 2.1795E-07 | 3.33 | 2.1840E-07 | 3.35 |
| Solutions for $v$ | | | | | | | | | | | | |
| 4 | 1.0799E-02 | | 1.5865E-02 | | 1.6451E-02 | | 2.1600E-03 | | 2.3325E-03 | | 2.8958E-03 | |
| 8 | 2.6363E-03 | 2.03 | 1.8037E-03 | 3.13 | 1.9330E-03 | 3.08 | 2.8422E-04 | 2.93 | 1.6816E-04 | 3.79 | 1.9644E-04 | 3.89 |
| 16 | 3.5676E-04 | 2.89 | 3.8329E-04 | 2.23 | 3.9504E-04 | 2.29 | 3.6252E-05 | 3.44 | 1.9610E-05 | 3.10 | 2.0508E-05 | 3.26 |
| 32 | 6.2368E-05 | 2.52 | 8.0461E-05 | 2.52 | 8.1200E-05 | 2.28 | 2.4576E-06 | 3.42 | 2.1967E-06 | 3.15 | 2.2208E-06 | 3.20 |
| 64 | 1.0004E-05 | 2.64 | 1.5355E-05 | 2.38 | 1.5323E-05 | 2.41 | 2.1893E-07 | 3.49 | 2.1795E-07 | 3.33 | 2.1840E-07 | 3.35 |

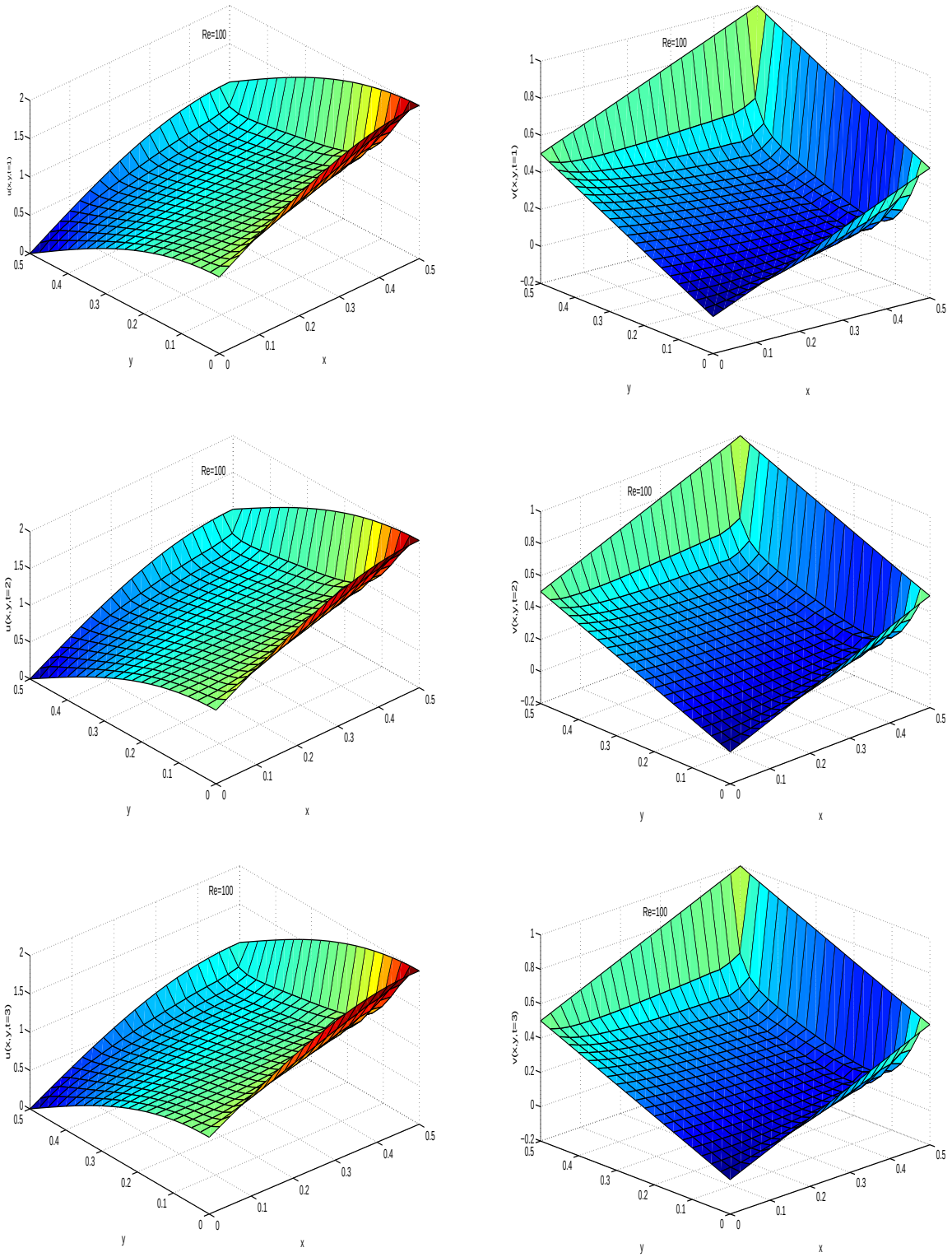


Figure 3.6: Behavior of MTB-DQM solutions of Example 3.4.3 for $Re = 100, h_x = h_y = 0.025$ and $\triangle t = 10^{-4}$ at $t = 3$

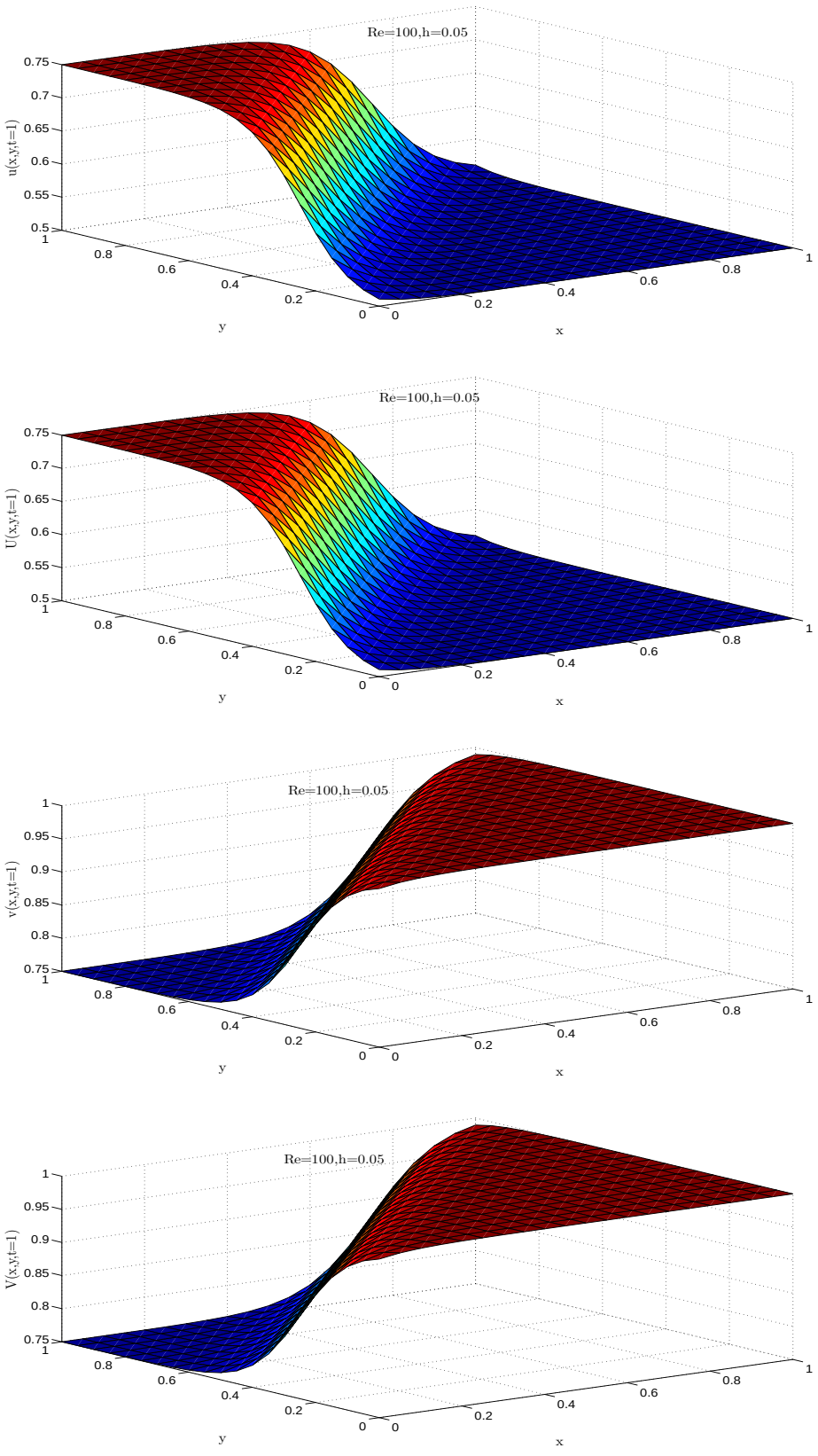


Figure 3.7: Comparison of MTB-DQM solution $(u, v)$ with exact solution $(U, V)$ of the velocity profile of Example 3.4.4 at $t = 1$ for $\nu = 0.01, h = 0.05, \triangle t = 0.0001$

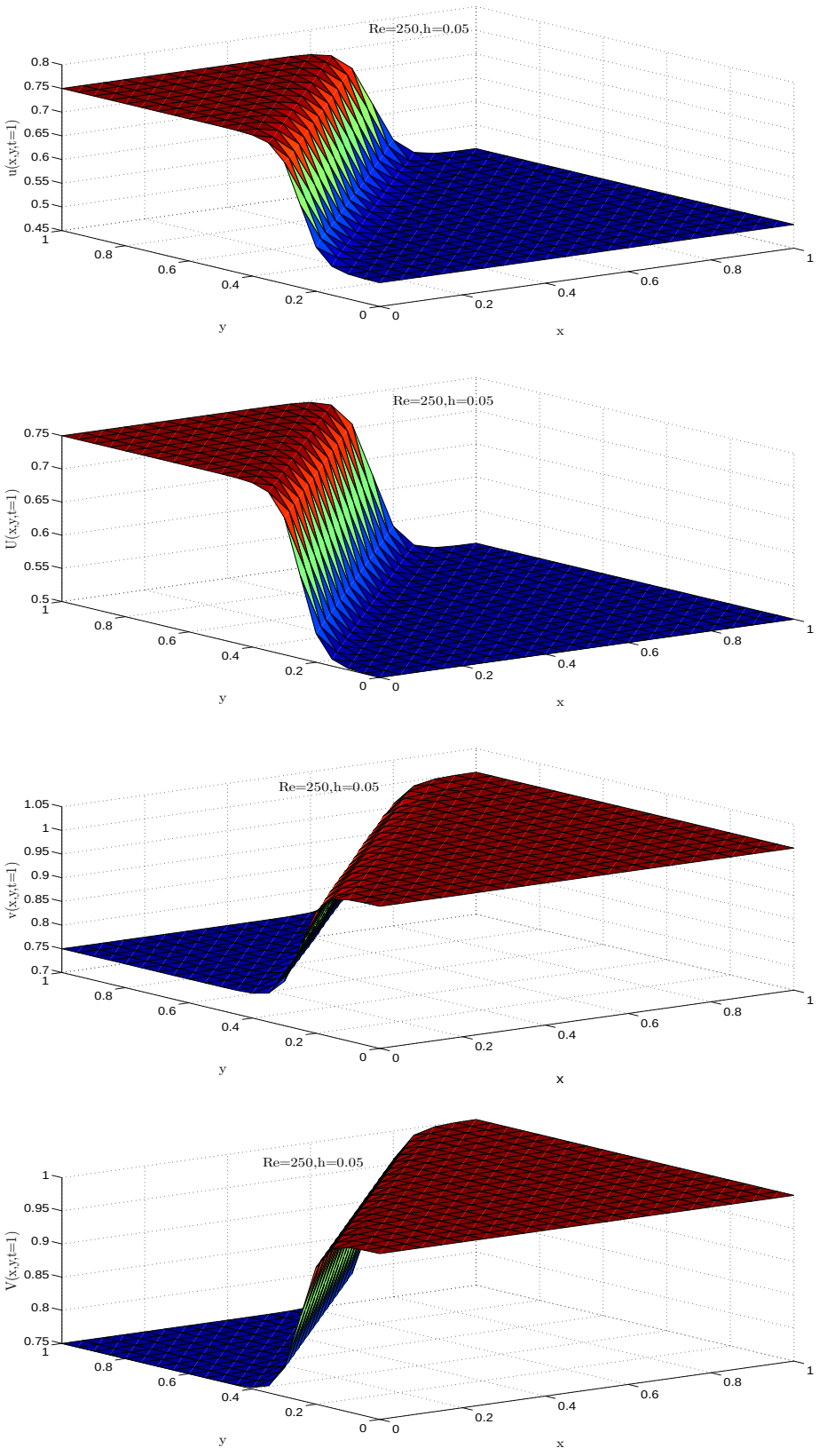


Figure 3.8: Comparison of MTB-DQM solution $(u, v)$ with exact solution $(U, V)$ of the velocity profile of Example 3.4.4 at $t = 1$ for $\nu = 0.004, h = 0.05, \triangle t = 0.0001$

## 3.5 Conclusion

In this chapter, a new approach "modified trigonometric cubic B-spline DQM" has been developed for numerical computation of nonlinear PDEs. Specially, the MTB-DQM has been implemented for both $(1+1)$D and $(2+1)$D coupled Burgers' equation. The weighting coefficients of the first-order derivative are obtained by using MTB-DQM while Shu's recursive formula based on polynomial DQM has been adopted to evaluate the weighting coefficients of second order derivatives. Then, the set of time-dependent first order ODEs obtained by using MTB-DQM to Burgers' equation in space, is solved via SSP-RK54 algorithm.

The findings show that the computational cost of both MCB-DQM [13] and MTB-DQM is same and the computed solutions agreed are well with the exact solutions as compared to the other schemes in $(1+1)$D coupled viscous Burgers' equation and are comparable to mECDQ [248] and Exp-MCB-DQM [268] for $(2+1)$D coupled viscous Burgers' equation. For the stability of the proposed method, the matrix stability analysis has been performed for various grid values, which concludes that the proposed method is stable for the nonlinear coupled viscous Burgers' equation in both one space dimension and two space dimension. Moreover, in the proposed method the solutions of Burgers' equation are obtained without any linearization technique and transformation.

# Chapter 4

# Numerical computation of Kuramoto-Sivashinsky equation via fourth order compact finite difference scheme

## 4.1 Introduction

The *Kuramoto-Sivashinsky* equation (4.1.1) was originally derived in the context of plasma instabilities, flame front propagation, and phase turbulence in the reaction-diffusion system [198]. *Kuramoto-Sivashinsky* equation models the fluctuations of the position of a flame front, the motion of a fluid going down a vertical wall, or a spatially uniform oscillating chemical reaction in a homogeneous medium [54]. This equation exhibits a chaotic behavior having a solution like traveling waves over a finite spatial domain moving without change of shape.

This chapter is concerned with the numerical simulation of one-dimensional *Kuramoto-Sivashinsky* equation subject to the initial and boundary conditions as follows

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \alpha\frac{\partial^2 u}{\partial x^2} + \gamma\frac{\partial^4 u}{\partial x^4} = 0, \tag{4.1.1}$$

$$u(x,0) = g(x),\ x \in \mathbb{R} \tag{4.1.2}$$

$$u(a,t) = \psi_1(t),\ u(b,t) = \psi_2(t),\ \frac{\partial u(a,t)}{\partial x} = \frac{\partial u(b,t)}{\partial x} = \frac{\partial^2 u(a,t)}{\partial x^2} = \frac{\partial^2 u(b,t)}{\partial x^2} = 0, \tag{4.1.3}$$

where $\mathbb{R}$ is the set of real numbers, $f, g, \psi_1, \psi_2$ are known functions. The nonlinear term in KSE counterbalance the dispersion term while dissipation terms show a mechanism for energy transfer. This equation has many applications in a variety of physical phenomena, e.g, in reaction-diffusion systems [144], long waves on the interface between two viscous fluids [97], hydrodynamics thin films [266] and flame front instability [253].

In the recent years, many researchers have developed the vigorous numerical techniques for numerical computation of time-dependent PDEs [149, 284, 285]. KSE studied numerically via several techniques, among others, Chebyshev spectral collocation methods [113], local discontinuous Galerkin methods [284], tanh function method [150], homotopy analysis method [146], inverse scattering method [59], homogeneous balance method [64], cubic B-spline finite difference collocation method [149], quintic B-spline collocation method (QBSC) [167], septic B-spline collocation method (SBSC) [295], higher-order finite element approach [16], finite difference discretization [10], fourth-order singly diagonally implicit Runge-Kutta method [58], lattice Boltzmann method (LBM) [285]. For more schemes, we refer the readers to [73, 94, 241, 248] and the references therein.

The compact finite difference scheme (CFDS) has been implemented for numerical simulation of various types of partial differential equations [152] such as hyperbolic equations [274], Navier-Stokes equation [213, 270], reaction diffusion [273], equation and Schrödinger equation [56]. A sixth-order compact finite difference has been developed for solving numerically to Helmholtz equation [264], advection diffusion equation [80], integro - differential equations [296], Burgers' equation [215], Burger - Fisher equation [217], Fisher's equation [23], Sine - Gordon equation [216], etc.

In this chapter, a new numeric solution of KSE is proposed via CFDS in space with the time integration SSP-RK43 algorithm [254]. First, CFDS is performed to convert KSE into a system of first-order ODEs, in time. Thereafter, time integration SSP-RK43 algorithm [254] is used to solve the resulting system of ODEs. To demonstrate the accuracy and utility of the proposed scheme, six test problems considered by many researchers are studied numerically.

## 4.2 Compact Finite Difference Schemes (CFDS)

In order to get the numerical solution of KSE (4.1.1) by CFDS, the spatial and time discretization is needed. The computational domain $\Omega_1 = \{x \in \mathbb{R} : a \leq x \leq b\}$ is portioned with uniform grid points: $a = x_1 < x_2, \ldots, x_{N-1} < x_N = b$ with $h = x_{i+1} - x_i = \frac{b-a}{N-1}$, $i = 1, 2, \ldots, N-1$.

The procedure of finding the spatial derivatives via CFDS is as follows:

### 4.2.1 First-order spatial derivative

The first-order derivative $i$-th collocation point can be given as

$$\theta u'_{i-1} + u'_i + \theta u'_{i+1} = b\frac{u_{i+2} - u_{i-2}}{4h} + a\frac{u_{i+1} - u_{i-1}}{2h} \tag{4.2.1}$$

In particular, for $a = \frac{2}{3}(2+\theta), b = \frac{1}{3}(4\theta - 1)$, Eq. (4.2.1) get reduced into a $\theta$-family of fourth-order tridiagonal schemes. Further, for $\theta = \frac{1}{3}$ the above scheme (4.2.1) is reduced in to the *sixth-order tridiagonal scheme* as

$$u'_{i-1} + 3u'_i + u'_{i+1} = \frac{1}{12h}(u_{i+2} + 28u_{i+1} - 28u_{i-1} - u_{i-2}), \tag{4.2.2}$$

with the truncation error $\frac{4}{7!}h^6 u_i^{(7)}$ [152].

The approximation for the derivatives at the boundary points $x_1$, $x_2$, $x_{N-1}$ and $x_N$, derived from one-sided schemes [215], respectively are

$$\begin{aligned}
u'_1 + 5u'_2 &= \frac{1}{60h}\left(-197u_1 - 25u_2 + 300u_3 - 100u_4 + 25u_5 - 3u_6\right)\\
2u'_1 + 11u'_2 + 2u'_3 &= \frac{1}{12h}\left(-80u_1 - 35u_2 + 136u_3 - 28u_4 + 8u_5 - u_6\right)\\
2u'_{N-2} + 11u'_{N-1} + 2u'_N &= \frac{1}{12h}\left(80u_N + 35u_{N-1} - 136u_{N-2} + 28u_{N-3} - 8u_{N-4} + u_{N-5}\right)\\
5u'_{N-1} + u'_N &= \frac{1}{60h}\left(197u_N + 25u_{N-1} - 300u_{N-2} + 100u_{N-3} - 25u_{N-4} + 3u_{N-5}\right)
\end{aligned}$$

The matrix representation of the scheme is given as

$$\begin{bmatrix} 1 & 5 & & & & & & \\ 2 & 11 & 2 & & & & & \\ & 1 & 3 & 1 & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & 1 & 3 & 1 & & \\ & & & & 2 & 11 & 2 & \\ & & & & & 5 & 1 & \end{bmatrix} \begin{bmatrix} u'_1 \\ u'_2 \\ \vdots \\ u'_i \\ \vdots \\ u'_{N-1} \\ u'_N \end{bmatrix} = \begin{bmatrix} \psi_1(u) \\ \psi_2(u) \\ \vdots \\ \psi_i(u) \\ \vdots \\ \psi_{N-1}(u) \\ \psi_N(u) \end{bmatrix} \tag{4.2.3}$$

where $u = (u_1, u_2, \ldots, u_N)^T$ and

$$\begin{aligned}
\psi_1(u) &= \frac{1}{60h}\left(-197u_1 - 25u_2 + 300u_3 - 100u_4 + 25u_5 - 3u_6\right) \\
\psi_2(u) &= \frac{1}{12h}\left(-80u_1 - 35u_2 + 136u_3 - 28u_4 + 8u_5 - u_6\right) \\
\psi_i(u) &= \frac{1}{12h}\left(u_{i+2} + 28u_{i+1} - 28u_{i-1} - u_{i-2}\right), \qquad i = 3, 4, \ldots, N-2, \\
\psi_{N-1}(u) &= \frac{1}{12h}\left(80u_N + 35u_{N-1} - 136u_{N-2} + 28u_{N-3} - 8u_{N-4} + u_{N-5}\right), \text{ and} \\
\psi_N(u) &= \frac{1}{60h}\left(197u_N + 25u_{N-1} - 300u_{N-2} + 100u_{N-3} - 25u_{N-4} + 3u_{N-5}\right)
\end{aligned}$$

### 4.2.2 Higher-order spatial derivative

The second-order derivative at $i$th collocation point can be given as follow

$$\theta u''_{i-1} + u''_i + \theta u''_{i+1} = b\frac{u_{i+2} - 2u_i + u_{i-2}}{4h^2} + a\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}. \tag{4.2.4}$$

In particular, for $a = \frac{4}{3}(1-\theta), b = \frac{1}{3}(-1+10\theta)$, Eq. (4.2.4) get reduced into a $\theta$-family of fourth-order tridiagonal scheme. For $\theta = \frac{1}{10}$, Eq. (4.2.4) reduced to

$$u''_{i-1} + 10u''_i + u''_{i+1} = \frac{12}{h^2}(u_{i+1} - 2u_i + u_{i-1}) \tag{4.2.5}$$

At the boundary nodes: $x_1$, $x_N$, the approximation formulae for the second-order derivatives are derived from Taylor series expansion about the nodes [23, 296] as follows:

$$10u_1'' + u_2'' = \frac{12}{h^2}\left(\frac{115}{36}u_1 - \frac{1555}{144}u_2 + \frac{89}{6}u_3 - \frac{773}{72}u_4 + \frac{151}{36}u_5 - \frac{11}{16}u_6\right)$$
$$u_{N-1}'' + 10u_N'' = \frac{12}{h^2}\left(-\frac{11}{16}u_{N-5} + \frac{151}{36}u_{N-4} - \frac{773}{72}u_{N-3} + \frac{89}{6}u_{N-2} - \frac{1555}{144}u_{N-1} + \frac{115}{36}u_N\right)$$

The above scheme for second order approximation can be written in matrix form as follows

$$\begin{bmatrix} 10 & 1 & & & & \\ 1 & 10 & 1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & 1 & 10 & 1 & \\ & & & 1 & 10 & \end{bmatrix} \begin{bmatrix} u_1'' \\ u_2'' \\ \vdots \\ u_{N-1}'' \\ u_N'' \end{bmatrix} = \begin{bmatrix} \phi_1(u) \\ \phi_2(u) \\ \vdots \\ \phi_{N-1}(u) \\ \phi_N(u) \end{bmatrix} \tag{4.2.6}$$

where

$$\phi_1(u) = \frac{12}{h^2}\left(\frac{115}{36}u_1 - \frac{1555}{144}u_2 + \frac{89}{6}u_3 - \frac{773}{72}u_4 + \frac{151}{36}u_5 - \frac{11}{16}u_6\right)$$
$$\phi_i(u) = \frac{12}{h^2}(u_{i+1} - 2u_i + u_{i-1}), \qquad i = 2, 3, \ldots, N-1, \text{ and}$$
$$\phi_N(u) = \frac{12}{h^2}\left(-\frac{11}{16}u_{N-5} + \frac{151}{36}u_{N-4} - \frac{773}{72}u_{N-3} + \frac{89}{6}u_{N-2} - \frac{1555}{144}u_{N-1} + \frac{115}{36}u_N\right).$$

The fourth-order spatial derivative is obtained by replacing $u$ by $u''$ in (4.2.6):

$$Au^{iv} = \phi(u''), \tag{4.2.7}$$

where $\phi(u'') = (\phi_1(u''), \phi_2(u''), \ldots, \phi_N(u''))$ and $A$ is the coefficient matrix defined in Eq. (4.2.6). The spatial derivatives every grid point are obtained by solving the above tridiagonal system of linear equations using "Thomas algorithm".

## 4.3 Implementation of the method

On putting the values of spatial derivatives approximated by CFDS, Eq. (4.1.1) is reduced into a system of first-order ODEs:

$$\frac{du_i}{dt} = L(u_i),\ i \in \{1, \ldots, N\}, \tag{4.3.1}$$

with initial condition $(*)$ and boundary conditions $(**)$, where $L \rightarrow$nonlinear differential operator defined by

$$L(u_i) = -u(x_i)u'(x_i) - \alpha u''(x_i) - \gamma u^{iv}(x_i).$$

Eq. (4.3.1) is solved by SSP-RK43 scheme through the following steps:

$$\begin{aligned}
u^{(1)} &= u^m + \frac{\triangle t}{2} L(u^m), \\
u^{(2)} &= u^{(1)} + \frac{\triangle t}{2} L(u^{(1)}), \\
u^{(3)} &= \frac{2}{3}u^m + \frac{u^{(2)}}{3} + \frac{\triangle t}{6} L(u^{(2)}), \\
u^{m+1} &= u^{(3)} + \frac{\triangle t}{2} L(u^{(3)}).
\end{aligned}$$

and thus, the solutions of $u(x,t)$ at the required time level are obtained.

## 4.4 Application of CFDS for KS equation

This section presents the numerical computation of six problems by adopting CFDS. The accuracy and efficiency of the method is measured by evaluating the $L_2$, $L_\infty$ error norms and the following global relative (GRE) error norm for the considered test problems.

$$\text{GRE} = \frac{\left(\sum_j |u_j - u_j^*|\right)}{\left(\sum_j |u_j^*|\right)}, \tag{4.4.1}$$

where $u_j \rightarrow$numerical solution, $u_j^* \rightarrow$analytical solution at grid $j$.

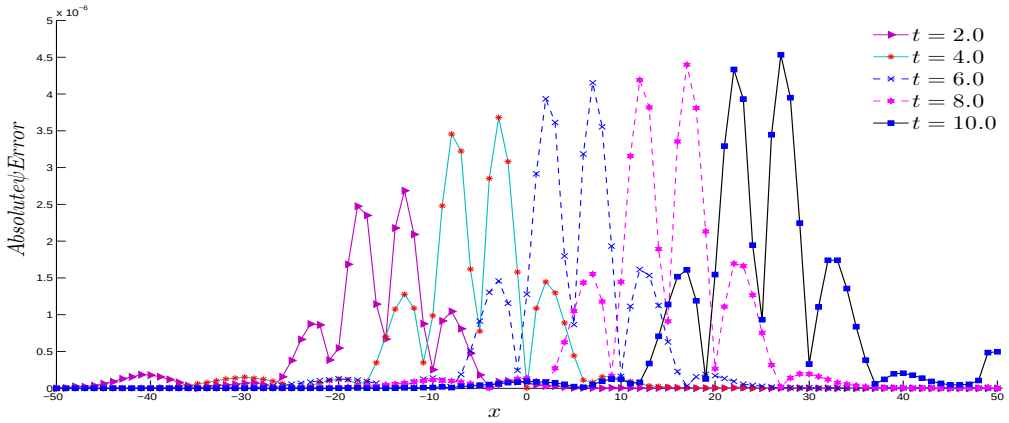


Figure 4.1: The absolute errors in Example 4.4.1 with $\Omega_1 = [-50, 50]$ with $\triangle t = 0.01$ and $N = 101$ at different time levels $t \leq 10$

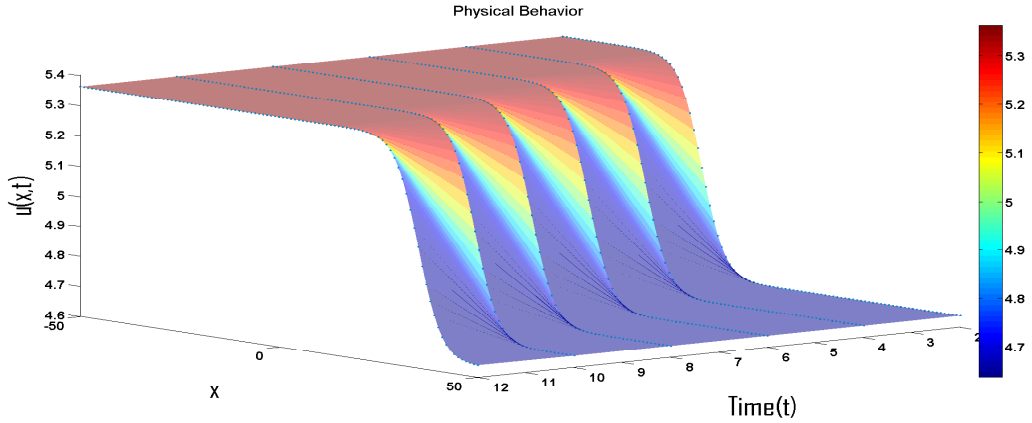


Figure 4.2: Physical behavior of Example 4.4.1 with $\Omega_1 = [-50, 50]$ with $\triangle t = 0.01$ and $N = 101$ at different time levels $t \leq 10$

Table 4.1: Comparison of GRE at different time levels t$\leq$ 12 with the errors due to earlier schemes [167], LBM [148], and SBSC [295] with $\triangle t = 0.01$

| $t$ | QBSC $N = 200$ | LBM $N = 200$ | SBSC $N = 200$ | CFDS $N = 100$ |
|---|---|---|---|---|
| 6 | 6.50927E-06 | 7.8808E-06 | 1.62464E-07 | 8.43458E-08 |
| 8 | 7.13154E-06 | 9.5324E-06 | 1.94032E-07 | 8.91175E-08 |
| 10 | 7.31029E-06 | 1.0891E-05 | 2.22878E-07 | 9.25444E-08 |
| 12 | 8.77659E-06 | 1.1793E-05 | 5.31428E-07 | 1.69186E-07 |

Table 4.2: Comparison of $L_\infty$ errors at time $t = 1, 2$ taking different grid pints and $\triangle t = 0.01$

| $N$ | 15 | 30 | 60 | 120 |
|---|---|---|---|---|
| $t = 1$ | 2.98E-01 | 9.83E-04 | 1.61E-05 | 8.20E-07 |
| $t = 2$ | 9.33E-01 | 5.96E-03 | 2.45E-05 | 1.32E-06 |

Table 4.3: Comparison of GRE, $L_\infty$ errors at different time levels $t \leq 5$ for $N = 141, \triangle t = 0.01$

| scheme | errors | $t = 1$ | $t = 2$ | $t = 3$ | $t = 4$ | $t = 5$ |
|---|---|---|---|---|---|---|
| CFDS | GRE | 9.5984E-09 | 1.4368E-08 | 1.7535E-08 | 1.9749E-08 | 2.2235E-08 |
| | $L_\infty$ | 4.6014E-07 | 6.6612E-07 | 8.1212E-07 | 9.0924E-07 | 9.7622E-07 |
| SBSC | GRE | 2.9543E-08 | 4.9686E-08 | 6.7009E-08 | 8.2345E-08 | 9.6082E-08 |
| | $L_\infty$ | 3.1791E-06 | 5.9019E-06 | 8.2807E-06 | 1.0385E-05 | 1.2264E-05 |

Table 4.4: Comparison of the errors at different time levels $t \leq 12$

| | $t = 1$ | $t = 2$ | $t = 3$ | $t = 4$ | $t = 6$ | $t = 8$ | $t = 10$ | $t = 12$ |
|---|---|---|---|---|---|---|---|---|
| | $N = 101, \triangle t = 0.01$ | | | | | | | |
| $L_\infty$ | 1.727E-06 | 2.685E-06 | 3.283E-06 | 3.681E-06 | 4.153E-06 | 4.399E-06 | 4.534E-06 | 3.040E-05 |
| $L_2$ | 3.961E-06 | 6.265E-06 | 7.764E-06 | 8.791E-06 | 1.005E-05 | 1.073E-05 | 1.114E-05 | 3.356E-05 |
| GRE | 3.415E-08 | 5.358E-08 | 6.609E-08 | 7.438E-08 | 8.435E-08 | 8.912E-08 | 9.254E-08 | 1.692E-07 |
| | $N = 101, \triangle t = 0.1$ | | | | | | | |
| $L_\infty$ | 1.097E-05 | 1.990E-05 | 2.727E-05 | 3.343E-05 | 4.306E-05 | 5.014E-05 | 5.545E-05 | 2.052E-04 |
| $L_2$ | 2.997E-05 | 5.451E-05 | 7.504E-05 | 9.240E-05 | 1.199E-04 | 1.404E-04 | 1.559E-04 | 2.709E-04 |
| GRE | 2.824E-07 | 5.050E-07 | 6.886E-07 | 8.440E-07 | 1.087E-06 | 1.266E-06 | 1.401E-06 | 1.939E-06 |
| | $N = 51, \triangle t = 0.1$ | | | | | | | |
| $L_\infty$ | 3.029E-05 | 5.117E-05 | 5.532E-05 | 6.668E-05 | 7.263E-05 | 7.497E-05 | 7.577E-05 | 7.614E-05 |
| $L_2$ | 7.465E-05 | 1.147E-04 | 1.391E-04 | 1.550E-04 | 1.740E-04 | 1.849E-04 | 1.920E-04 | 1.971E-04 |
| GRE | 6.759E-07 | 9.793E-07 | 1.255E-06 | 1.367E-06 | 1.540E-06 | 1.629E-06 | 1.693E-06 | 1.634E-06 |

**Example 4.4.1.** *The first test case deals with KSE* (4.1.1) *with* $\alpha = -1,\ \gamma = 1$ *as follows*

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} + \frac{\partial^4 u}{\partial x^4} = 0, t > 0.$$

*The initial and boundary conditions are extracted from the following exact solution [167]:*

$$u(x,t) = \beta + \frac{15\tanh^3(k(x-\beta t - x_0)) - 45\tanh(k(x-\beta t - x_0))}{19^{3/2}}. \tag{4.4.2}$$

The numerical computation of Example 4.4.1 is performed for $\beta = 5, k = \frac{1}{2\sqrt{19}}$, $x_0 = -25$ with $N = 101, 141$ and $\triangle t = 0.1, 0.01$. For $\triangle t = 0.01, N = 101$, the comparison of the computed GRE at different time levels $t \leq 12$ using CFDS with the errors from earlier schemes: LBM [148], QBSC [167] and SBSC [295], is reported in Table 4.1. $L_\infty$, global relative errors at different time levels $t \in \{1, 2, 3, 4, 5\}$ are compared with the errors computed by using SBSC [295] in Table 4.3. The errors are reported in 4.2 and Table 4.4 for different number of grids and time-steps. The findings show that the proposed solutions are more accurate than the solutions computed in SBSC [295], LBM [148] and QBSC [167]. The absolute errors and the physical behavior of the CFDS solutions at different time levels are depicted in Figure 4.1 and Figure 4.2, respectively.

**Example 4.4.2.** *This test case deals with KSE* (4.1.1) *with* $\alpha = 1, \gamma = 1$*, that is,*

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{\partial^2 u}{\partial x^2} + \frac{\partial^4 u}{\partial x^4} = 0, t > 0,$$

*subject to the initial and boundary conditions extracted form exact solutions* (4.4.3) *[148]:*

$$u(x,t) = \beta + \frac{15}{19}\sqrt{\frac{11}{19}}\left[-9\tanh(k(x-\beta t - x_0)) + 11\tanh^3(k(x-\beta t - x_0))\right]. \tag{4.4.3}$$

Table 4.5: Comparison of GRE with errors in LBM [148], QBSC [167] for $\triangle t = 0.01$ at different time levels $t \leq 4$

| scheme | $N$ | $t = 1$ | $t = 2$ | $t = 3$ | $t = 4$ |
|---|---|---|---|---|---|
| QBSC | 150 | 3.82E-04 | 5.51E-04 | 7.04E-04 | 8.64E-04 |
| LBM | 150 | 6.79E-04 | 1.15E-03 | 1.59E-03 | 2.01E-03 |
| CFDS | 63 | 4.33E-04 | 5.79E-04 | 8.74E-04 | 3.57E-03 |

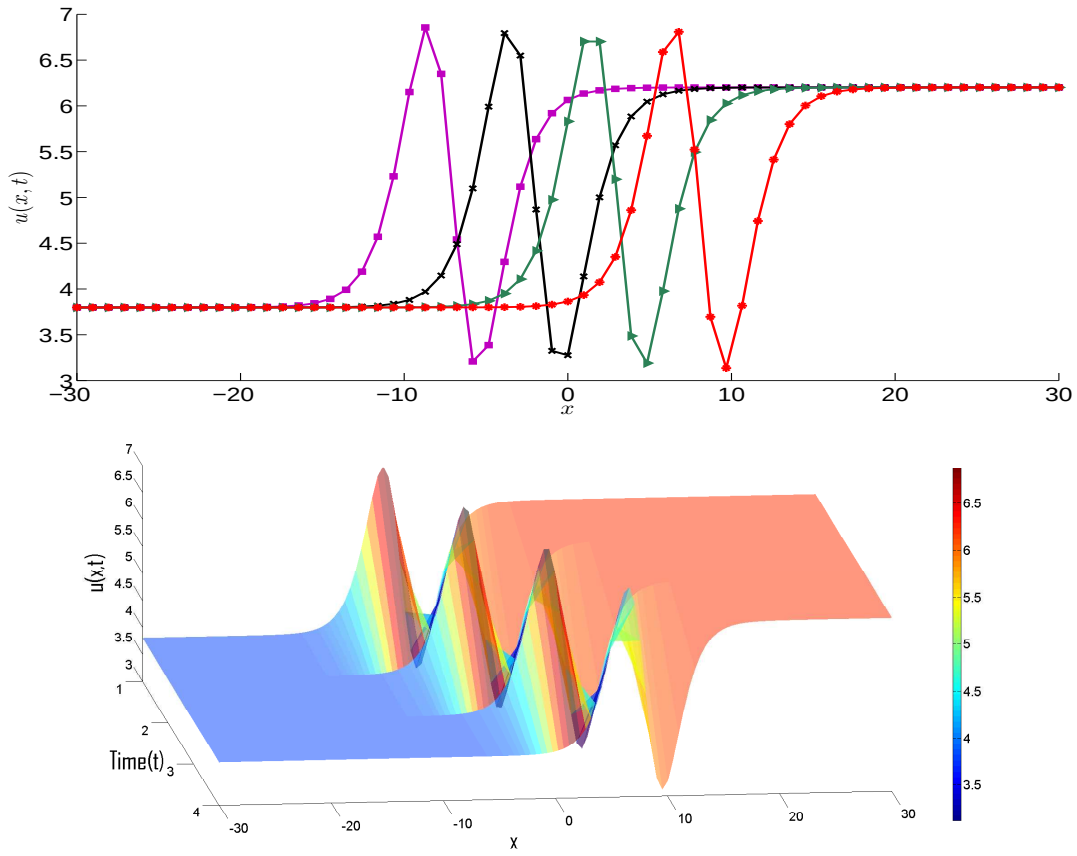


Figure 4.3: Solution behavior of Example 4.4.2 in $\Omega_1 = [-30, 30]$ at different time levels $t \leq 4$ in two and three-dimensions with $\triangle t = 0.01, N = 63$

*The numerical computation of Example 4.4.2 is performed for* $\beta = 5, k = 0.5\sqrt{\frac{11}{19}}$ *and* $x_0 = -12$. *The comparison of the GREs computed via CFDS at different time levels* $(t \leq 4)$ *for* $\triangle t = 0.01, N = 63$ *with the errors computed in LBM [148] and QBSC [167] is reported in Table 4.5. The findings show that the proposed results are comparable to the results in [148, 167]. The numerical results with analytical solutions for* $\Omega_1 = [-30, 30]$ *at different time levels are depicted in Figure 4.3.*

**Example 4.4.3.** *KSE* (4.1.1) *with* $\alpha = -\nu, \gamma = 0$ *get reduced to Burgers' equation*

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = 0. \tag{4.4.4}$$

*This test case deals with Burgers' equation* (4.4.4) *in* $\Omega = [0, 1.2]$ *together with the initial*

Table 4.6: Comparison of CFDS solutions of Example 4.4.3 with MCB-DQM [13], Mittal and Jain [171], Shu et al. [227] and the exact solutions for $\nu = 0.005$

| $x$ | $t$ | [227] with $h = 10^{-4}$ | | [171] | [13] | CFDS | Exact Value |
|---|---|---|---|---|---|---|---|
| | | $\beta = 1$ | $\beta = 0.5$ | $h = 0.005$ | $h = 0.01$ | $h = 0.01$ | |
| | | $\triangle t = 0.01$ | $\triangle t = 0.01$ | $\triangle t = 10^{-3}$ | $\triangle t = 0.01$ | $\triangle t = 0.01$ | |
| 0.20 | 1.7 | 0.1176565 | 0.1174841 | 0.1176452 | 0.1176450 | 0.1176450 | 0.1176452 |
| | 2.5 | 0.0800527 | 0.0798389 | 0.0799990 | 0.0799989 | 0.0799990 | 0.0799990 |
| | 3.0 | 0.0667147 | 0.0665176 | 0.0666658 | 0.0666658 | 0.0666658 | 0.0666658 |
| | 3.5 | 0.0571820 | 0.0570060 | 0.0571422 | 0.0571422 | 0.0571422 | 0.0571422 |
| 0.40 | 1.7 | 0.2332111 | 0.2348504 | 0.2351690 | 0.2351680 | 0.2351680 | 0.2351677 |
| | 2.5 | 0.1591735 | 0.1596608 | 0.1599771 | 0.1599770 | 0.1599770 | 0.1599769 |
| | 3.0 | 0.1328314 | 0.1330273 | 0.1333211 | 0.1333210 | 0.1333210 | 0.1333209 |
| | 3.5 | 0.1139606 | 0.1140077 | 0.1142780 | 0.1142780 | 0.1142780 | 0.1142779 |
| 0.6 | 1.7 | 0.2940048 | 0.2961269 | 0.2958570 | 0.2959160 | 0.2959090 | 0.2959097 |
| | 2.5 | 0.2347876 | 0.2376699 | 0.2381299 | 0.2381200 | 0.2381210 | 0.2381207 |
| | 3.0 | 0.1973222 | 0.1990478 | 0.1994839 | 0.1994800 | 0.1994810 | 0.1994805 |
| | 3.5 | 0.1697753 | 0.1708231 | 0.1712257 | 0.1712240 | 0.1712240 | 0.1712242 |
| 0.8 | 1.7 | 0.0008917 | 0.0006640 | 0.0006381 | 0.0006464 | 0.0006464 | 0.0006465 |
| | 2.5 | 0.1103866 | 0.1036067 | 0.1021325 | 0.1020930 | 0.1020940 | 0.1020957 |
| | 3.0 | 0.2088346 | 0.2093735 | 0.2088032 | 0.2088380 | 0.2088360 | 0.2088359 |
| | 3.5 | 0.2119293 | 0.2143409 | 0.2145938 | 0.2145870 | 0.2145870 | 0.2145869 |

*condition considered at* $t = 1$ *as follows*

$$u(x, t = 1) = x \left[1 + \exp\left(\frac{1}{4\nu}\left(x^2 - \frac{1}{4}\right)\right)\right]^{-1},$$

*and the boundary conditions*

$$u(0, t) = u(1.2, t) = 0, t > 1.$$

*The exact solution of for this example is given by*

$$u(x, t) = \frac{x}{t}\left(1 + \left(\frac{t}{t_0}\right)^{1/2} \exp\left(\frac{x^2}{4\nu t}\right)\right)^{-1}, \; t_0 = \exp\left(1/8\nu\right), \; \textit{for } t \geq 1.$$

The comparison of computed CFDS solutions at different time levels are presented in

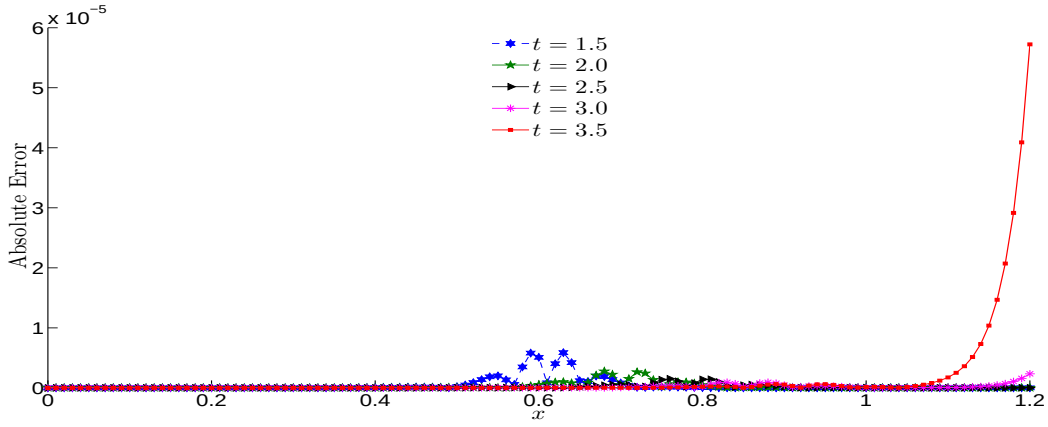


Figure 4.4: The absolute errors and the physical behavior of Example 4.4.3 at different time levels $t \leq 3.5$

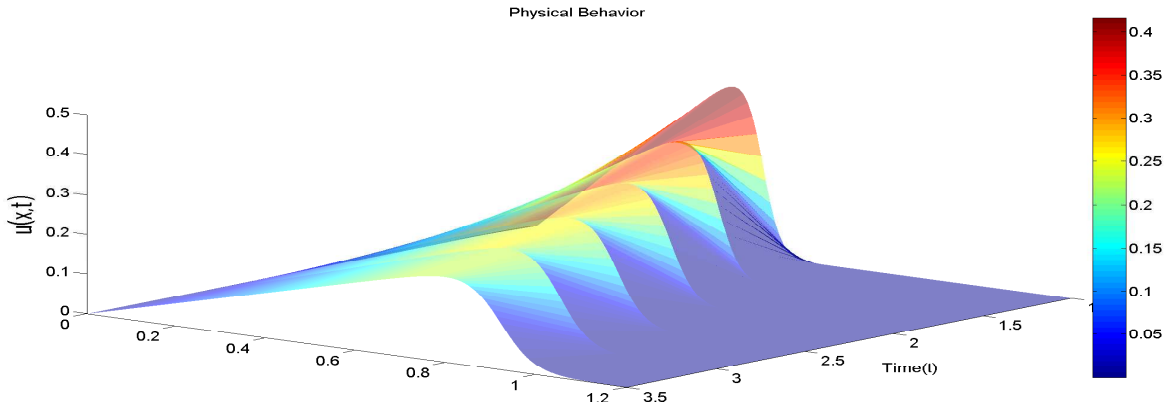


Figure 4.5: The absolute errors and the physical behavior of Example 4.4.3 at different time levels $t \leq 3.5$

Table 4.6 for selected node points, with parameters $\nu = 0.005$, $h = 0.01$ and $\triangle t = 0.01$. The comparison is done with solutions obtained by Mittal and Jain [171], Shu et al. [227], Arora-Singh [13], and the exact solutions. In Table 4.7, $L_2$ and $L_\infty$ errors for $\nu = 0.005$ at different time levels, $t \leq 3.5$, are compared with the errors computed by using several earlier schemes. At $t = 3.6$, $L_2$ and $L_\infty$ errors are $10^{-5}$ and $7 \times 10^{-5}$, respectively which are same as in MCB-DQM [13] and are very less in comparison to the errors computed by three methods proposed in [128]. These findings show that CFDS produces more accurate numeric solutions which are better than almost all the earlier schemes, and approaching the exact solutions. The absolute errors and physical behavior behaviors of this example are depicted in Figure 4.4 and Figure 4.5, respectively.

Table 4.7: Comparison of $L_2$ and $L_\infty$ errors in the CFDS solutions of Example 4.4.3 for $\nu = 0.005$ at different time levels $t \leq 3.5$

| | | | $t = 1.7$ | | $t = 2.4$ | | $t = 3.1$ | | $t = 3.25$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| Methods | $N$ | $\triangle t$ | $L_2 \times 10^3$ | $L_\infty \times 10^3$ | $L_2 \times 10^3$ | $L_\infty \times 10^3$ | $L_2 \times 10^3$ | $L_\infty \times 10^3$ | $L_2 \times 10^3$ | $L_\infty \times 10^3$ |
| CFDS | 121 | 0.01 | 0.00095 | 0.00428 | 0.00041 | 0.00167 | 0.00051 | 0.00330 | 0.001284 | 0.009148 |
| MCB-DQM [13] | 121 | 0.01 | 0.00191 | 0.00777 | 0.00086 | 0.00308 | 0.00065 | 0.00331 | 0.001341 | 0.00918 |
| QRTDQ [133] | 101 | 0.001 | 0.109 | 0.434 | 0.100 | 0.339 | 0.091 | 0.266 | | |
| CBCDQ [124] | 101 | 0.001 | | | 0.210 | 0.680 | 0.190 | 0.530 | | |
| BSFEM [49] | 50 | 0.1 | 0.857 | 2.576 | 0.423 | 1.242 | 0.230 | 0.680 | | |
| CSC [210] | 50 | 0.1 | 0.857 | 2.576 | 0.423 | 1.242 | 0.235 | 0.688 | | |
| Galerkin [297] | 200 | 0.01 | 0.857 | 2.576 | 0.423 | 1.242 | 0.235 | 0.688 | | |
| QBCM1 [154] | 200 | 0.01 | 0.017 | 0.061 | 0.012 | 0.058 | 0.601 | 4.434 | | |
| QBCM2 [154] | 200 | 0.01 | 0.358 | 1.211 | 0.251 | 0.807 | 0.630 | 4.790 | | |
| PDQ [123] | 200 | 0.01 | 0.015 | 0.056 | 0.011 | 0.064 | 0.584 | 4.301 | | |
| | | | | | $t = 2.5$ | | | | | |
| QBCM [55] | 200 | 0.01 | 0.072 | 0.311 | 0.051 | 0.189 | | | 1.129 | 8.983 |
| CBCM [55] | 200 | 0.01 | 2.466 | 27.577 | 2.111 | 25.15 | | | 1.925 | 21.084 |
| QRKM [55] | 200 | 0.01 | 0.026 | 0.091 | 0.031 | 0.115 | | | 1.111 | 8.000 |
| | | | | | | | $t = 3.00$ | | $t = 3.50$ | |
| CFDS | 121 | 0.01 | 0.00095 | 0.00428 | 0.00038 | 0.00152 | 0.00033 | 0.00158 | 0.005815 | 0.04090 |
| MCB-DQM [13] | 121 | 0.01 | 0.00191 | 0.00777 | 0.00778 | 0.00275 | 0.00056 | 0.0017 | 0.006177 | 0.04335 |
| MCB-CM [171] | 241 | 0.01 | 0.0252 | 0.0994 | 0.0151 | 0.0549 | 0.0118 | 0.0414 | 0.0117 | 0.0486 |
| [227]$\beta = 0.5$ | 12001 | 0.01 | 0.38421 | 1.34728 | 0.49135 | 1.55470 | 0.51508 | 1.5529 | 0.525855 | 1.52196 |
| [227]$\beta = 1$ | 12001 | 0.01 | 3.08966 | 10.4040 | 2.72048 | 8.29747 | 2.39922 | 6.9880 | 2.12110 | 5.94321 |

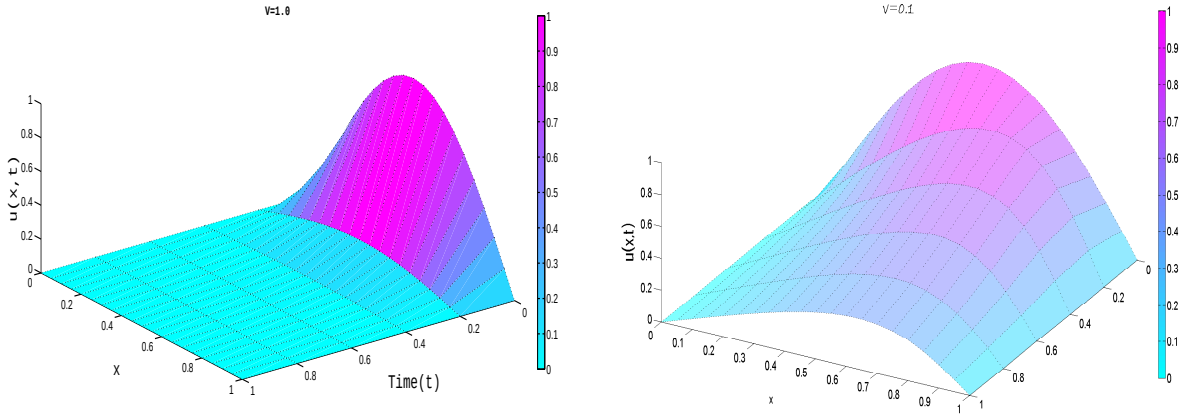


Figure 4.6: CFSDS solution behavior of Example 4.4.4 for $\nu = 1.0, \nu = 0.1$ at different time levels ($t \leq 1$) with $N = 21, \triangle t = 0.001$

**Example 4.4.4.** *This test case deals with KSE* (4.1.1) *with* $\alpha = 1$, *and* $\gamma = 0$ *(i.e., Burgers' equation* (4.4.4)*) in* $\Omega_1 = [0, 1]$ *with initial condition* $u(x, 0) = \sin(\pi x)$, *and boundary conditions* $u(0, t) = u(1, t) = 0$. *The analytical solution of this problem is given by*

$$u(x,t) = \frac{4\pi\nu \sum_{j=1}^{\infty} jI_j\left(\frac{1}{2\pi\nu}\right) \sin(j\pi x)\, exp\left(-j^2\pi^2\nu t\right)}{I_0\left(\frac{1}{2\pi\nu}\right) + 2\sum_{j=1}^{\infty} I_j\left(\frac{1}{2\pi\nu}\right) \cos(j\pi x)\, exp\left(-j^2\pi^2\nu t\right)}, \tag{4.4.5}$$

*where* $I_j$ *are the modified Bessel's functions.*

The numerical solutions are obtained for different values of $\nu$ as follows:

**a)** The numerical solutions are computed at different time levels with parameter values $\nu = 1.0, h = 0.05$ and $\triangle t = 0.001$ at some selected node points. Table 4.8 presents the comparison of CFDS numerical solutions with the exact solutions and the solutions computed by using several schemes in [13, 55, 134, 167, 171]. The findings show that CFDS produces comparable results.

**b)** In Table 4.9, the CFDS solutions are computed at different time levels with parameter values $\nu = 0.1, h = 0.025, \triangle t = 0.004$. The findings from Table 4.9, concludes the CFDS solutions are comparably much better than the numerical solutions computed by using the earlier schemes in [13, 55, 134, 167, 171, 215], and approaching towards exact solutions.

The physical behavior of the solution for $\nu = 0.1, 1.0$ is depicted in Figure 4.6.

**Example 4.4.5.** *This test case deals with KSE* (4.1.1) *with* $\alpha = 1$ *and* $\gamma = 0.5$ *in the domain* $[-10, 10]$. *The numerical solutions are computed for the initial and boundary con-*

Table 4.8: Comparison of CFDS solutions of Example 4.4.4 for $\nu = 1.0$ with earlier schemes, and exact solutions with parameter $h = 0.05$ and $\triangle t = 0.001$

| $x$ | | [55] | [171] | [134] | [13] | [167] | CFDS | Exact |
|---|---|---|---|---|---|---|---|---|
| | $\triangle t \rightarrow$ | $10^{-4}$ | 0.00025 | 0.000125 | 0.00025 | 0.001 | 0.001 | |
| | $t$ | 0.0125 | 0.025 | 0.025 | 0.025 | 0.004 | 0.05 | $\leftarrow h$ |
| 0.25 | 0.4 | 0.01357 | 0.01354 | 0.01363 | 0.0135710 | - - | - - | 0.01357 |
| | 0.6 | 0.00189 | 0.00188 | 0.00190 | 0.0018888 | 0.00185 | 0.001889 | 0.00189 |
| | 0.8 | 0.00026 | 0.00026 | 0.00026 | 0.0002624 | 0.00026 | 0.000262 | 0.00026 |
| | 1.0 | 0.00004 | 0.00004 | 0.00003 | 0.0000365 | 0.00004 | 0.000036 | 0.00004 |
| 0.50 | 0.4 | 0.01923 | 0.01920 | 0.01932 | 0.0192336 | - - | - - | 0.01923 |
| | 0.6 | 0.00267 | 0.00266 | 0.00269 | 0.0026719 | 0.00262 | 0.002672 | 0.00267 |
| | 0.8 | 0.00037 | 0.00037 | 0.00037 | 0.0003712 | 0.00036 | 0.000371 | 0.00037 |
| | 1.0 | 0.00005 | 0.00005 | 0.00005 | 0.0000516 | 0.00005 | 0.000051 | 0.00005 |
| 0.75 | 0.4 | 0.01362 | 0.01360 | 0.01369 | 0.0136298 | - - | - - | 0.01363 |
| | 0.6 | 0.00189 | 0.00188 | 0.00190 | 0.0018899 | 0.00185 | 0.001890 | 0.00189 |
| | 0.8 | 0.00026 | 0.00026 | 0.00026 | 0.0002625 | 0.00026 | 0.000262 | 0.00026 |
| | 1.0 | 0.00004 | 0.00004 | 0.00003 | 0.0000365 | 0.00004 | 0.000036 | 0.00004 |

*ditions extracted from the following exact solutions [149]*

$$u(x,t) = -\frac{1}{10k} + \frac{60}{19}k\left(\alpha - 38\gamma k^2\right)\tanh\left(kx + \frac{t}{10}\right) + 120\gamma k^3 \tanh^3\left(kx + \frac{t}{10}\right), \quad (4.4.6)$$

*where* $k = \frac{1}{2}\sqrt{\frac{11\alpha}{19\gamma}}$.

The errors are found in Table 4.10 are comparable with that the errors computed via a method based on the finite difference and collocation method using B-splines by Lakestani and Dehghan [149]. The numerical solutions are also presented graphically in Figure 4.7 at different time levels $1 \leq t \leq 5$, which shows the same characteristics as depicted in [149].

**Example 4.4.6.** *Consider the KSE* (4.1.1) *with* $\alpha = 1$ *and* $\gamma = 1$ *in the domain* $[-30, 30]$ *which results in the nonlinear partial differential equation exhibiting the chaotic behavior over a finite spatial domain. The solution is obtained for the Gaussian initial condition [149, 167, 272]:* $u(x,0) = \exp(-x^2)$ *with the BCs* $u(-30,t) = u(30,t) = 0$. *The numerical solutions at different time levels* $5 \leq t \leq 30$ *are presented graphically in Fig 4.8, and it is found that the solutions depict the same characteristics as in it were in [149, 167, 272].*

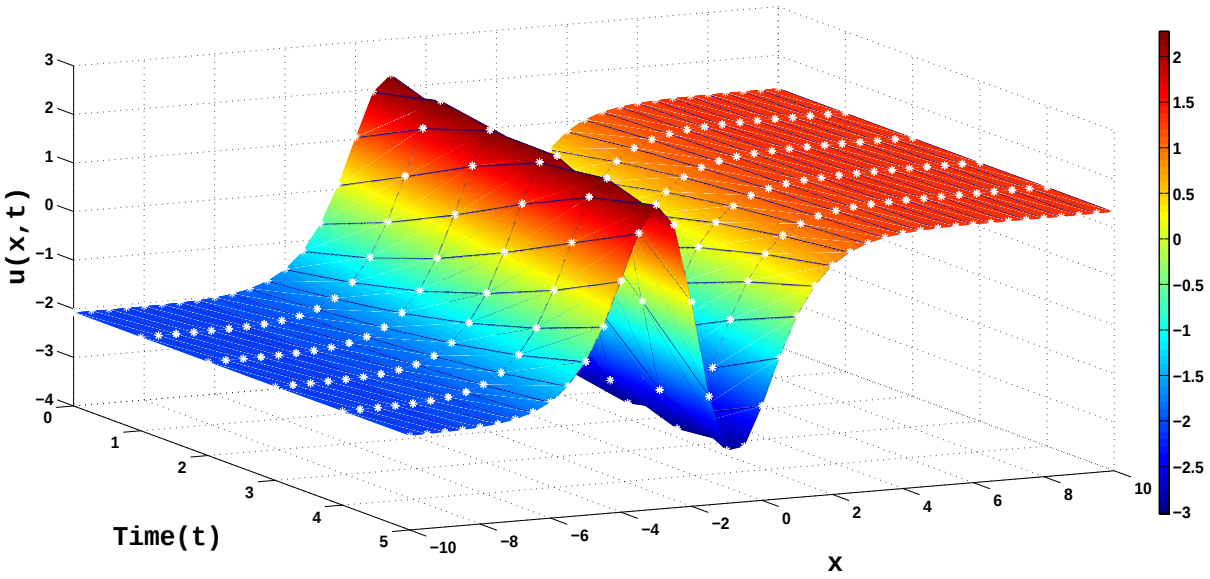


Figure 4.7: Physical behavior of numerical solutions of Example 4.4.5 at $t \leq 5$ with $N = 41, \triangle t = 0.01$

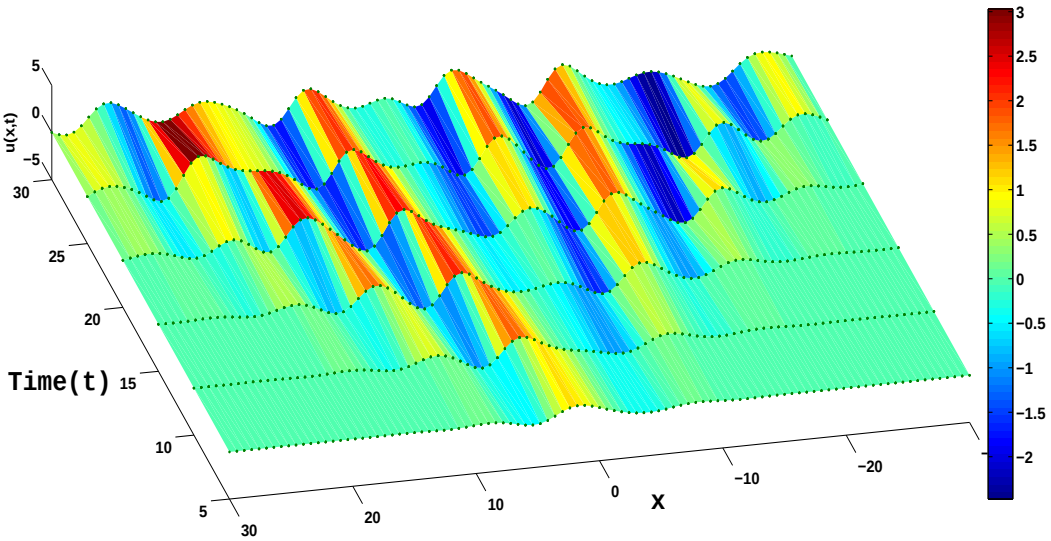


Figure 4.8: Physical behavior of numeric solutions of Example 4.4.6 at $t \leq 5$ with $N = 101, \triangle t = 0.01$

Table 4.9: Comparison of CFDS solutions of Example 4.4.4 for $\nu = 0.1$ with solutions by earlier schemes, and exact solutions

| $x$ | | [55] | [171] | [134] | [13] | [167] | CFDS | [215] | Exact |
|---|---|---|---|---|---|---|---|---|---|
| $\triangle t_{\rightarrow}$ | | $10^{-4}$ | 0.0025 | 0.00125 | 0.004 | 0.0001 | 0.001 | 0.00001 | |
| | $t$ | 0.0125 | 0.025 | 0.025 | 0.025 | 0.004 | 0.025 | 0.0125 | $\leftarrow h$ |
| 0.25 | 0.4 | 0.30890 | 0.30892 | 0.30910 | 0.3089280 | | 0.308894 | 0.308894 | 0.30889 |
| | 0.6 | 0.24075 | 0.24077 | 0.24093 | 0.2407550 | 0.24073 | 0.240739 | 0.240739 | 0.24074 |
| | 0.8 | 0.19569 | 0.19572 | 0.19586 | 0.1956840 | 0.19519 | 0.195676 | 0.195676 | 0.19568 |
| | 1.0 | 0.16258 | 0.16261 | 0.16274 | 0.1625700 | 0.16206 | 0.162565 | 0.162565 | 0.16256 |
| | 3.0 | 0.02720 | 0.02718 | 0.02720 | 0.0272047 | 0.02699 | 0.027202 | – | 0.02720 |
| 0.50 | 0.4 | 0.56965 | 0.56970 | 0.56973 | 0.5696530 | – | 0.569629 | 0.569632 | 0.56963 |
| | 0.6 | 0.44723 | 0.44729 | 0.44736 | 0.4472170 | 0.44683 | 0.447205 | 0.447206 | 0.44721 |
| | 0.8 | 0.35925 | 0.35930 | 0.35943 | 0.3592450 | 0.35875 | 0.359235 | 0.359236 | 0.35924 |
| | 1.0 | 0.29192 | 0.29195 | 0.29213 | 0.2919250 | 0.29136 | 0.291915 | 0.291916 | 0.29192 |
| | 3.0 | 0.04019 | 0.04016 | 0.04032 | 0.0402085 | 0.03993 | 0.040205 | – | 0.04021 |
| 0.75 | 0.4 | 0.62538 | 0.62520 | 0.62573 | 0.6253490 | – | 0.625361 | 0.625438 | 0.62544 |
| | 0.6 | 0.48715 | 0.48694 | 0.48760 | 0.4872040 | 0.48613 | 0.487212 | 0.487215 | 0.48721 |
| | 0.8 | 0.37385 | 0.37365 | 0.37434 | 0.3739350 | 0.37281 | 0.373920 | 0.373922 | 0.37392 |
| | 1.0 | 0.28741 | 0.28724 | 0.28788 | 0.2874930 | 0.28642 | 0.287473 | 0.287474 | 0.28747 |
| | 3.0 | 0.02976 | 0.02974 | 0.29881 | 0.0297753 | 0.02952 | 0.029772 | – | 0.02977 |

Table 4.10: Comparison of the errors in Example 4.4.5 for $t = 1$ with the parameter $\triangle t = 0.01$ and $N = 40$

| Errors | CFDS | Lakestani & Dehghan [149] | |
|---|---|---|---|
| | | $(j = 3)$ | $(j = 4)$ |
| $L_\infty$ | 8.58E-04 | 4.80E-03 | 2.10E-03 |
| $L_2$ | 1.44E-03 | 4.40E-03 | 1.60E-03 |

Table 4.11: The errors in Example 4.4.5 at different time levels $t \leq 5$ with the parameters $\triangle t = 0.01$ and $N = 40$

| Errors | $t = 0.5$ | $t = 1$ | $t = 2$ | $t = 3$ | $t = 4$ | $t = 5$ |
|---|---|---|---|---|---|---|
| $L_\infty$ | 8.07E-04 | 8.58E-04 | 1.01E-03 | 1.24E-03 | 1.31E-03 | 1.32E-03 |
| $L_2$ | 1.29E-03 | 1.44E-03 | 1.94E-03 | 2.45E-03 | 2.84E-03 | 3.01E-03 |
| GRE | 1.17E-04 | 1.46E-04 | 2.19E-04 | 3.11E-04 | 3.97E-04 | 4.25E-04 |

## 4.5 Conclusion

This chapter deals with a new numerical simulation of the *Kuramoto-Sivashinsky* equation, which is approximated by using compact finite difference scheme (CFDS). The CFDS is used to convert KSE into a system of first-order ODEs, in time. The SSP-RK43 algorithm

is adopted to solve the resulting system of ODEs. Six test problems considered by many researchers have been studied to confirm the accuracy and utility of the proposed scheme. The findings can be summarized as follows

- ✓ The proposed results are found to be in good agreement with the exact solutions. The effects of space and time resolutions on the accuracy of the present scheme are also investigated through detailed simulations.

- ✓ The advantage of this scheme is that it is easy to implement to a (non)linear equation and easy to program. It requires less number of grid points, and so, it leads to the least accumulation of numerical errors.

- ✓ The approximate solutions have been computed without any transformation or linearization. Easy, computational efficiency and economical implementation is the strength of this scheme.

- ✓ This scheme can be adopted as a promising technique to solve multi-dimensional problems arising in various engineering and physical modeling problems.

# Chapter 5

# An approximate series solutions of time fractional Navier- Stokes equation by two reliable methods

## 5.1 Introduction

The most famous governing differential equation of motion of viscus fluid flow, Navier-Stokes equation (regarded as Newton's second law of motion for fluid substances) was derived in 1822 [180]. This equation is a mixture of momentum equation, the equation of continuity (i.e., expresses the conservation of momentum and mass). This equation describes many physical phenomena such as ocean currents, liquid flow in pipes, blood flow, air flow around the wings of an aircraft. The fractional modeling of Navier-Stokes equation was first done in 2005 by El-Shahed and Salem [60]. The following is the multi-dimensional, time-fractional of Navier- Stokes equation for the incompressible fluid flow of kinematic viscosity $\nu = \eta/\rho$ and constant density $\rho$ as given in [34]:

$$\begin{cases} \mathcal{D}_t^{\alpha} U + (U \cdot \nabla)U = \nu \nabla^2 U - \frac{1}{\rho}\nabla p, & \text{on } \Omega \times (0,T) \\ \nabla \cdot U = 0, & \text{on } \Omega \times (0,T) \\ U = 0, & \text{on } \partial\Omega \times (0,T) \end{cases} \tag{5.1.1}$$

where $U = (u, v, w)$ is the fluid vector, $p$ is the pressure and $u(x, y, z, t), v(x, y, z, t)$ and $w(x, y, z, t)$ are the velocity components in the direction of $x, y$ and $z$ at time $t$, respectively. $\partial\Omega$ is the boundary of the spatial domain $\Omega$, $\eta$ is the dynamic viscosity, the ratio $\nu = \eta/\rho$ be the kinematic viscosity of the flow and $\rho$ is the density.
Further, if $p$ is known, then $g_1 = -\frac{1}{\rho}\frac{\partial p}{\partial x}$, $g_2 = -\frac{1}{\rho}\frac{\partial p}{\partial y}$, $g_3 = -\frac{1}{\rho}\frac{\partial p}{\partial z}$ can be determined. Time fractional Navier - Stokes equation (5.1.1) in cartesian coordinates reduces to

$$\begin{cases} \mathcal{D}_t^\alpha u + u\dfrac{\partial u}{\partial x} + v\dfrac{\partial u}{\partial y} + w\dfrac{\partial u}{\partial z} = \nu\left(\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} + \dfrac{\partial^2 u}{\partial z^2}\right) + g_1, \\ \mathcal{D}_t^\alpha v + u\dfrac{\partial v}{\partial x} + v\dfrac{\partial v}{\partial y} + w\dfrac{\partial v}{\partial z} = \nu\left(\dfrac{\partial^2 v}{\partial x^2} + \dfrac{\partial^2 v}{\partial y^2} + \dfrac{\partial^2 v}{\partial z^2}\right) + g_2, \\ \mathcal{D}_t^\alpha w + u\dfrac{\partial w}{\partial x} + v\dfrac{\partial w}{\partial y} + w\dfrac{\partial w}{\partial z} = \nu\left(\dfrac{\partial^2 w}{\partial x^2} + \dfrac{\partial^2 w}{\partial y^2} + \dfrac{\partial^2 w}{\partial z^2}\right) + g_3, \end{cases} \tag{5.1.2}$$

El-Shahed and Salem presented the first approximate analytical solutions of time fractional Navier-Stokes equation via Laplace transform, finite Hankel transforms and finite Fourier sine transform [60]. After this seminal work, the nonlinear fractional Navier - Stokes equation has been studied by many techniques such as adomaian decomposition method (ADM) [34,177], Homotopy analysis method (HAM) [68,200], residual power series method [100] and a new method based on operational matrices [238].

Integral transforms have capability to transform differential equations into algebraic equations, which allows simple and systematic solution procedure for nonlinear PDEs [173, 249]. Keeping this in mind various semi-analytical hybrid methods have been developed to compute an approximate solution of fractional Navier-Stokes equation, among them, homotopy perturbation transform method (HPTM) [139,250], mixture of ADM and Laplace transform algorithm (LTA) [136], the coupling of LTA and finite Hankel transform [41], new iterative Elzaki transform method [276] and many more.

This chapter deals with the approximate series solutions of multi-dimensional, time-fractional of Navier- Stokes equation (5.1.1) with Caputo type fractional derivative when $p$ is known. The proposed solutions are obtained by two reliable methods: 1) fractional reduced differential transform method (FRDTM), 2) New integral projected differential transform method (NIPDTM). The description of these methods is given in the following

## 5.2 Description of the semi-analytical methods

The NIPDTM is a hybrid semi-analytical method, in which projected differential transform together with new integral transform [105] is adopted for the analytical study of models consisting of time-fractional PDEs.

### 5.2.1 New integral projected differential transform method

Let $\mathcal{F}$ be the set of functions of exponential order as defined below:

$$\mathcal{F} = \left\{ f(t) : \exists M, k_1, k_2 > 0 \text{ such that } |f(t)| \leq M e^{\frac{|t|}{k_i^2}} \text{ whenever } t \in (-1)^i \times [0, \infty) \right\},$$

where a given $f \in \mathcal{F}$ the constant $M$ must be finite while $k_1, k_2$ may be finite or infinite.

**Definition 5.2.1** ( [105,106,218]). *The new integral transform $\mathcal{K}\{u(t)\} = \mathcal{U}(\mu)$ of $u(t) \in \mathcal{F}$ is defined by*

$$\mathcal{U}(\mu) = \mathcal{K}\{u(t)\} = \frac{1}{\mu} \int_0^\infty e^{\frac{-t}{\mu^2}} u(t) dt \tag{5.2.1}$$

The following are some properties of new integral transform

**Theorem 5.2.2** ( [105, 106, 218]). ***a)*** *The new integral transform of $\frac{\partial^n u}{\partial t^n}$ is given by*

$$\mathcal{K}\left\{ \frac{\partial^n u(x,t)}{\partial t^n} \right\} = \frac{\mathcal{U}(x,\mu)}{\mu^{2n}} - \sum_{k=0}^{n-1} \frac{1}{\mu^{2(n-k)-1}} \frac{\partial^k u(x,0^+)}{\partial t^k}, \quad n \geq 1. \tag{5.2.2}$$

***b)*** *The new integral transform of the Caputo fractional derivative $D_t^\alpha u(x,t)$ and Riemann-Liouville fractional integral $\mathcal{J}_t^\alpha u(x,t)$ of $u(x,t)$ is given by*

*(i)* $\mathcal{K}\{\mathcal{J}_t^\alpha u(x,t)\} = \mu^{2\alpha} \mathcal{U}(x,\mu)$,

*(ii)* $\mathcal{K}\{\mathcal{D}_t^\alpha u(x,t)\} = \frac{\mathcal{U}(x,\mu)}{\mu^{2\alpha}} - \sum_{k=0}^{n-1} \frac{1}{\mu^{2(\alpha-k)-1}} \frac{\partial^k u(x,0^+)}{\partial t^k}, \quad n-1 < \alpha \leq n \in \mathbb{N},$

***c)*** $\mathcal{K}\left\{ \frac{t^{n\alpha}}{\Gamma(1+n\alpha)} \right\} = \mu^{2n\alpha+1}$.

The readers are referred to [219, 220] for more properties of new integral transform.

**A. Implementation of NIPDTM method for fractional PDEs**

Consider the following initial value system of time fractional nonlinear PDEs

$$\begin{cases} D_t^{\alpha} u(x,t) + L[u(x,t)] + N[u(x,t)] = g(x,t) > 0, x \in \mathbb{R}, \quad 0 < \alpha \leq 1, \\ u(x,0) = h(x), \end{cases} \tag{5.2.3}$$

where $D_t^{\alpha} u$ Caputo-fractional derivative of $u$ of order $\alpha$, $L \rightarrow$linear operator, $N \rightarrow$nonlinear operator and $g(x,t) \rightarrow$ smooth functions.

The new integral transform on (5.2.3) yields

$$\mathcal{K}\left\{D_t^{\alpha} u(x,t)\right\} + \mathcal{K}\left\{L[u(x,t)]\right\} + \mathcal{K}\left\{N[u(x,t)]\right\} = \mathcal{K}\left\{g(x,t)\right\}. \tag{5.2.4}$$

On using Theorem 5.2.2$(a,b)$, we get

$$\mathcal{U}(x,\mu) = \mu^{2\alpha}\mathcal{G}(x,\mu) + \mu h(x) - \mu^{2\alpha}\left[\mathcal{K}\left\{L[u(x,t)]\right\} + \mathcal{K}\left\{N[u(x,t)]\right\}\right]. \tag{5.2.5}$$

Next, inverse new integral transform on (5.2.5) yields

$$u(x,t) = g_1(x,t) - \mathcal{K}^{-1}\left\{\mu^{2\alpha}\left[\mathcal{K}\left\{L[u(x,t)]\right\} + \mathcal{K}\left\{N[u(x,t)]\right\}\right]\right\} \tag{5.2.6}$$

where $g_1(x,t) = h(x) + \mathcal{K}^{-1}\left\{\mu^{2\alpha}\mathcal{G}(x,\mu)\right\}$, i.e., the term arising from the initial conditions and source terms.

The projected differential transform (PDT) method on equation (5.2.6) yields the following recurrence relation

$$\begin{cases} U(x,0) = g_1(x,t,), \\ U(x,h+1) = -\mathcal{K}^{-1}\left\{\mu^{2\alpha}\mathcal{K}[A_h + B_h]\right\}, \ h = 0,1,2,\ldots. \end{cases} \tag{5.2.7}$$

where $A_h = \ \mathrm{PDT}\left\{N[u(x,t)]\right\}, B_h = \ \mathrm{PDT}\left\{L[u(x,t)]\right\}$.

The above expression (5.2.7) is referred to as "new integral-projected differential transform (NIPDT)" of the problem (5.2.3).

The approximation solution of (5.2.3) is given by

$$u(x,t) = \sum_{h=0}^{\infty} U(x,h). \tag{5.2.8}$$

The advantage of NIPDTM is its capability of computing approximate solutions PDEs in forms of a fast convergent series [105]. The implementation of NIPDTM for time fractional Navier- Stokes equation is described in the following

**B. NIPDT for time fractional Navier- Stokes equation**

The NIPDT on Equation (5.1.2) yields

$$\begin{cases} U(X,h+1) = \mathcal{K}^{-1}\left(\mu^{2\alpha}\mathcal{K}\left[A_h^{(1)} + B_h^{(1)}\right]\right), \quad U(X,0) = u_0(X) + \dfrac{g_1 t^{\alpha}}{\Gamma(1+\alpha)} \\ V(X,h+1) = \mathcal{K}^{-1}\left(\mu^{2\alpha}\mathcal{K}\left[A_h^{(2)} + B_h^{(2)}\right]\right), \quad V(X,0) = v_0(X) + \dfrac{g_2 t^{\alpha}}{\Gamma(1+\alpha)} \\ W(X,h+1) = \mathcal{K}^{-1}\left(\mu^{2\alpha}\mathcal{K}\left[A_h^{(3)} + B_h^{(3)}\right]\right), \quad W(X,0) = w_0(X) + \dfrac{g_3 t^{\alpha}}{\Gamma(1+\alpha)} \end{cases} \tag{5.2.9}$$

where

$$\begin{aligned} A_h^{(1)} &= -\sum_{m=0}^{h}\left(U(X,m)\frac{\partial U(X,h-m)}{\partial x} + V(X,m)\frac{\partial U(X,h-m)}{\partial y} + W(X,m)\frac{\partial U(X,h-m)}{\partial z}\right), \\ B_h^{(1)} &= \nu\left\{\frac{\partial^2 U(X,h)}{\partial x^2} + \frac{\partial^2 U(X,h)}{\partial y^2} + \frac{\partial^2 U(X,h)}{\partial z^2}\right\}, \\ A_h^{(2)} &= -\sum_{m=0}^{h}\left(U(X,m)\frac{\partial V(X,h-m)}{\partial x} + V(X,m)\frac{\partial V(X,h-m)}{\partial y} + W(X,m)\frac{\partial V(X,h-m)}{\partial z}\right), \\ B_h^{(2)} &= \nu\left\{\frac{\partial^2 V(X,h)}{\partial x^2} + \frac{\partial^2 V(X,h)}{\partial y^2} + \frac{\partial^2 V(X,h)}{\partial z^{(2)}}\right\}, \\ A_h^{(3)} &= -\sum_{m=0}^{h}\left(U(X,m)\frac{\partial W(X,h-m)}{\partial x} + V(X,m)\frac{\partial W(X,h-m)}{\partial y} + W(X,m)\frac{\partial W(X,h-m)}{\partial z}\right), \\ B_h^{(3)} &= \nu\left\{\frac{\partial^2 W(X,h)}{\partial x^2} + \frac{\partial^2 W(X,h)}{\partial y^2} + \frac{\partial^2 W(X,h)}{\partial z^2}\right\}, \quad h = 0,1,2\ldots \end{aligned} \tag{5.2.10}$$

Once the recurrence relation (5.2.9) solved for $U(X,h), V(X,h)$ and $W(X,h)$ $h \geq 0$, the approximate solution of time fractional Navier stokes equation (5.1.2) is given by

$$u(X,t) = \sum_{h=0}^{\infty} U(X,h), \quad v(X,t) = \sum_{h=0}^{\infty} V(X,h), \quad w(X,t) = \sum_{h=0}^{\infty} W(X,h). \tag{5.2.11}$$

### 5.2.2 FRDTM for time fractional Navier- Stokes equation

Keeping the properties of FRDTM in mind from Theorem 1.3.1 and Theorem 1.3.2, the FRDTM of Equation (5.1.2) with known $p$ yields the following

$$\begin{cases} \dfrac{\Gamma(1+(1+k)\alpha)}{\Gamma(1+k\alpha)} U_\alpha^{k+1} = \nu\nabla^2 U_\alpha^k - \displaystyle\sum_{\ell=0}^{k} \left( \frac{\partial U_\alpha^\ell}{\partial x} U_\alpha^{k-\ell} + \frac{\partial U_\alpha^\ell}{\partial y} V_\alpha^{k-\ell} + \frac{\partial U_\alpha^\ell}{\partial z} W_\alpha^{k-\ell} \right) + g_1\delta(k), \\ \dfrac{\Gamma(1+(1+k)\alpha)}{\Gamma(1+k\alpha)} V_\alpha^{k+1} = \nu\nabla^2 V_\alpha^k - \displaystyle\sum_{\ell=0}^{k} \left( \frac{\partial V_\alpha^\ell}{\partial x} U_\alpha^{k-\ell} + \frac{\partial V_\alpha^\ell}{\partial y} V_\alpha^{k-\ell} + \frac{\partial V_\alpha^\ell}{\partial z} W_\alpha^{k-\ell} \right) + g_2\delta(k), \\ \dfrac{\Gamma(1+(1+k)\alpha)}{\Gamma(1+k\alpha)} W_\alpha^{k+1} = \nu\nabla^2 W_\alpha^k - \displaystyle\sum_{\ell=0}^{k} \left( \frac{\partial W_\alpha^\ell}{\partial x} U_\alpha^{k-\ell} + \frac{\partial W_\alpha^\ell}{\partial y} V_\alpha^{k-\ell} + \frac{\partial W_\alpha^\ell}{\partial z} W_\alpha^{k-\ell} \right) + g_3\delta(k), \end{cases} \tag{5.2.12}$$

where $\nabla^2 \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$, $X = (x,y,z)$, $U_\alpha^k \equiv U_\alpha^k(X)$ etc.. One can obtain the recursive values of $U_\alpha^k, V_\alpha^k, W_\alpha^k$ $(k \geq 1)$ by solving above the system of recurrence relation simultaneously once the value $U_\alpha^0, V_\alpha^0, W_\alpha^0$ are known.

The approximate solution of time fractional Navier- Stokes equation (5.1.2) is obtained by taking inverse FRDTM of $U_\alpha^k, V_\alpha^k$ and $W_\alpha^k$'s, respectively

$$\begin{aligned} u(X,t) &= \sum_{k=0}^{\infty} U_\alpha^k \, (t)^{k\alpha}, \\ v(X,t) &= \sum_{k=0}^{\infty} V_\alpha^k \, (t)^{k\alpha}, \\ w(X,t) &= \sum_{k=0}^{\infty} W_\alpha^k \, (t)^{k\alpha}. \end{aligned}$$

The $m$th order solution of time fractional Navier- Stokes equation (5.1.2) can be read as

$$S_m(u) = \sum_{k=0}^{m} U_\alpha^k \, (t)^{k\alpha}, \quad S_m(v) = \sum_{k=0}^{m} V_\alpha^k \, (t)^{k\alpha}, \quad S_m(w) = \sum_{k=0}^{m} W_\alpha^k \, (t)^{k\alpha}.$$

## 5.3 Numerical result and discussion

This section deals with the main aim, to test the accuracy and efficiency of the two methods considering the following three test problems

**Example 5.3.1.** *This test case deals with two dimensional time-fractional Navier-Stokes equation with* $g_1 = -g_2 = g$ *as follows*

$$\begin{cases} \mathcal{D}_t^{\alpha} u + u\dfrac{\partial u}{\partial x} + u\dfrac{\partial v}{\partial y} = \nu\left(\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2}\right) + g, \\ \mathcal{D}_t^{\alpha} v + u\dfrac{\partial v}{\partial x} + v\dfrac{\partial v}{\partial y} = \nu\left(\dfrac{\partial^2 v}{\partial x^2} + \dfrac{\partial^2 v}{\partial y^2}\right) - g, \end{cases} \tag{5.3.1}$$

*subject to the initial condition*

$$u(X,0) = -\sin(x+y),\ v(X,0) = \sin(x+y),\ X = (x,y). \tag{5.3.2}$$

*The exact solution of Example 5.3.1 with* $\alpha = 1, g = 0$ *is given by*

$$u(x,y,t) = -e^{-2\nu t}\sin(x+y); \qquad v(x,y,t) = e^{-2\nu t}\sin(x+y). \tag{5.3.3}$$

**Example 5.3.2.** *This test case deals with two dimensional time-fractional Navier-Stokes equation* (5.3.1) *subject to the initial condition*

$$u(X,0) = -e^{x+y},\ v(X,0) = e^{x+y},\ X = (x,y) \tag{5.3.4}$$

*The exact solution of Example 5.3.2 with* $\alpha = 1, g = 0$ *[34] as given by*

$$u(x,y,t) = -e^{x+y+2\nu t}, \quad v(x,y,t) = e^{x+y+2\nu t}, \tag{5.3.5}$$

**Example 5.3.3.** *This test case deals with three dimensional time-fractional Navier-Stokes equation* (5.1.2) *with* $g_1 = g_2 = g_3 = 0$ *subject to the initial condition*

$$u(X,0) = -0.5x+y+z,\ v(X,0) = x-0.5y+z,\ w(X,0) = x+y-0.5z.\ X = (x,y,z) \tag{5.3.6}$$

*The exact solution of Example 5.3.3 with $\alpha = 1$ ( i.e., classical Navier-Stokes equation) subject to the initial conditions* (5.3.6) *is given by*

$$\left.\begin{aligned} u(X,t) &= \frac{-\frac{x}{2}+y+z-\frac{9xt}{4}}{1-\frac{9t^2}{4}}.\\ v(X,t) &= \frac{x-\frac{y}{2}+z-\frac{9yt}{4}}{1-\frac{9t^2}{4}}.\\ w(X,t) &= \frac{x+y-\frac{z}{2}-\frac{9zt}{4}}{1-\frac{9t^2}{4}}. \end{aligned}\right\} \tag{5.3.7}$$

### 5.3.1 Solution of Example 5.3.1

**A. By fractional reduced differential transform method**

The following recurrence relation is obtained by applying FRDTM on (5.3.1)-(5.3.2)

$$\left\{\begin{aligned} &\frac{\Gamma(1+(1+k)\alpha)}{\Gamma(1+k\alpha)}U_\alpha^{k+1}(X)+\sum_{\ell=0}^{k}\left(\frac{\partial U_\alpha^\ell(X)}{\partial x}U_\alpha^{k-\ell}(X)+\frac{\partial U_\alpha^\ell(X)}{\partial y}V_\alpha^{k-\ell}(X)\right)\\ &=\nu\nabla^2(U_\alpha^k(X))+g\delta(k), \quad U_\alpha^0(X)=-\sin(x+y),\\ &\frac{\Gamma(1+(1+k)\alpha)}{\Gamma(1+k\alpha)}V_\alpha^{k+1}(X)+\sum_{\ell=0}^{k}\left(\frac{\partial V_\alpha^\ell(X)}{\partial x}U_\alpha^{k-\ell}(X)+\frac{\partial V_\alpha^\ell(X)}{\partial y}V_\alpha^{k-\ell}(X)\right)\\ &=\nu\nabla^2(V_\alpha^k(X))-g\delta(k), \quad V_\alpha^0(X)=\sin(x+y),\ k\geq 0. \end{aligned}\right. \tag{5.3.8}$$

On solving the recurrence relation (5.3.8), we get

$$\begin{aligned} &U_\alpha^1(X)=\frac{2\nu}{\Gamma(1+\alpha)}\sin(x+y)+\frac{g}{\Gamma(1+\alpha)}; \quad && V_\alpha^1(X)=-\frac{2\nu}{\Gamma(1+\alpha)}\sin(x+y)-\frac{g}{\Gamma(1+\alpha)}\\ &U_\alpha^2(X)=-\frac{(2\nu)^2}{\Gamma(1+2\alpha)}\sin(x+y); && V_\alpha^2(x,y)=\frac{(2\nu)^2}{\Gamma(1+2\alpha)}\sin(x+y)\\ &\vdots \qquad \vdots \quad \vdots \qquad \vdots && \\ &U_\alpha^k(X)=-\frac{(-2\nu)^k}{\Gamma(1+k\alpha)}\sin(x+y); && V_\alpha^k(x,y)=\frac{(-2\nu)^k}{\Gamma(1+k\alpha)}\sin(x+y),\ k\geq 2, \end{aligned}$$

By using inverse FRDT, we have

$$\begin{aligned} u(x,y,t) &= \sum_{k=0}^{\infty} U_{\alpha}^{k}(X)t^{\alpha k} - \sin(x+y)\sum_{k=0}^{\infty}\frac{(-2\nu t^{\alpha})^{k}}{\Gamma(1+k\alpha)} + \frac{gt^{\alpha}}{\Gamma(1+\alpha)}, \\ &= -\sin(x+y)E_{\alpha,1}(-2\nu t^{\alpha}) + \frac{gt^{\alpha}}{\Gamma(1+\alpha)}, \end{aligned} \tag{5.3.9}$$

$$\begin{aligned} v(x,y,t) &= \sum_{k=0}^{\infty} V_{\alpha}^{k}(X)t^{\alpha k} = \sin(x+y)\sum_{k=0}^{\infty}\frac{(-2\nu t^{\alpha})^{k}}{\Gamma(1+k\alpha)} - \frac{gt^{\alpha}}{\Gamma(1+\alpha)}, \\ &= \sin(x+y)E_{\alpha,1}(-2\nu t^{\alpha}) - \frac{gt^{\alpha}}{\Gamma(1+\alpha)}. \end{aligned} \tag{5.3.10}$$

where $E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty}\frac{z^{k}}{\Gamma(\beta+k\alpha)}$, for $\alpha,\beta > 0$ denotes the Mittag-leffler function with two parameters [194], and so, $E_{1,1}(z) = e^{z}$.

In particular for $g = 0, \alpha = 1$, the solutions (5.3.9)-(5.3.10) is the closed form of the exact solution (5.3.3) of the classical Navier- Stokes equation for the velocity field.

**B. By new integral projected differential transform method**

Applying NIPDT on (5.3.1) with (5.3.2) yields the following recurrence relation

$$\begin{aligned} U(X,h+1) &= \mathcal{K}^{-1}\left(\mu^{2\alpha}\mathcal{K}\left[A_{h}^{(1)} + B_{h}^{(1)}\right]\right),\ U(X,0) = -\sin(x+y) + \frac{gt^{\alpha}}{\Gamma(1+\alpha)} \\ V(X,h+1) &= \mathcal{K}^{-1}\left(\mu^{2\alpha}\mathcal{K}\left[A_{h}^{(2)} + B_{h}^{(2)}\right]\right),\ V(X,0) = \sin(x+y) - \frac{gt^{\alpha}}{\Gamma(1+\alpha)}, \end{aligned} \tag{5.3.11}$$

where $h \geq = 0$ and

$$\begin{aligned} A_{h}^{(1)} &= -\sum_{m=0}^{h}\left(U(X,m)\frac{\partial U(X,h-m)}{\partial x} + V(X,m)\frac{\partial U(X,h-m)}{\partial y}\right), \\ A_{h}^{(2)} &= -\sum_{m=0}^{h}\left(U(X,m)\frac{\partial V(X,h-m)}{\partial x} + V(X,m)\frac{\partial V(X,h-m)}{\partial y}\right), \\ B_{h}^{(1)} &= \nu\left\{\frac{\partial^{2}U(X,h)}{\partial x^{2}} + \frac{\partial^{2}U(X,h)}{\partial y^{2}}\right\},\ B_{h}^{(2)} = \nu\left\{\frac{\partial^{2}V(X,h)}{\partial x^{2}} + \frac{\partial^{2}V(X,h)}{\partial y^{2}}\right\}. \end{aligned} \tag{5.3.12}$$

On solving the relation (5.3.11), we get

$$\begin{array}{lll}
A_0^{(1)}=0, & B_0^{(1)}=2\nu\sin(x+y), & U(X,1)=\frac{(2\nu t^\alpha)}{\Gamma(\alpha+1)}\sin(x+y),\\
A_0^{(2)}=0, & B_0^{(2)}=-2\nu\sin(x+y), & V(X,1)=-\frac{(2\nu t^\alpha)}{\Gamma(\alpha+1)}\sin(x+y),\\
A_1^{(1)}=0, & B_1^{(1)}=\frac{(-4\nu^2)t^\alpha}{\Gamma(1+\alpha)}\sin(x+y), & U(X,2)=-\frac{(2t^\alpha\nu)^2}{\Gamma(2\alpha+1)}\sin(x+y),\\
A_1^{(2)}=0, & B_1^{(2)}=\frac{(4\nu^2)t^\alpha}{\Gamma(1+\alpha)}\sin(x+y), & V(X,2)=\frac{(2t^\alpha\nu)^2}{\Gamma(2\alpha+1)}\sin(x+y),\\
A_2^{(1)}=0, & B_2^{(1)}=\frac{(8\nu^3t^{2\alpha})}{\Gamma(1+2\alpha)}\sin(x+y), & U(X,3)=\frac{(2\nu t^\alpha)^3}{\Gamma(3\alpha+1)}\sin(x+y),\\
A_2^{(2)}=0, & B_2^{(2)}=\frac{(-8\nu^2t^{2\alpha})}{\Gamma(1+2\alpha)}\sin(x+y), & V(X,3)=-\frac{(2t^\alpha\nu)^3}{\Gamma(3\alpha+1)}\sin(x+y),\\
\vdots & \vdots & \vdots
\end{array} \tag{5.3.13}$$

Hence the solutions $u(X,t),\ v(X,t)$ are obtained as

$$\begin{aligned}
u(X,t)&=\sum_{h=0}^{\infty}u(X,h)=-\sin(x+y)\sum_{h=0}^{\infty}\frac{(-2\nu t^\alpha)^h}{\Gamma(1+h\alpha)}+\frac{gt^\alpha}{\Gamma(1+\alpha)}\\
&=-\sin(x+y)E_{\alpha,1}(-2\nu t^\alpha)+\frac{gt^\alpha}{\Gamma(1+\alpha)},
\end{aligned} \tag{5.3.14}$$

$$\begin{aligned}
v(X,t)&=\sum_{h=0}^{\infty}V(X,h)=\sin(x+y)\sum_{h=0}^{\infty}\frac{(-2\nu t^\alpha)^h}{\Gamma(1+h\alpha)}-\frac{gt^\alpha}{\Gamma(1+\alpha)},\\
&=\sin(x+y)E_{\alpha,1}(-2\nu t^\alpha)-\frac{gt^\alpha}{\Gamma(1+\alpha)}.
\end{aligned} \tag{5.3.15}$$

In particular, $E_{1,1}(z)=e^z$. For $g=0,\alpha=1$ Eq. (5.3.14)-(5.3.15) is the closed form of the exact solution (5.3.3) of the classical Navier- Stokes equation for the velocity field.

**Remark 5.3.4.** *From the above study, we have found that both methods* FRDTM *and* NIPDTM *yield the same approximate solutions. The absolute errors in different order* $(m=4,6,8,10)$ *approximate solutions of velocity profile* $(u,v)$ *of Example 5.3.1 with different value of* $Re=1,10,100$ *is depicted in Figure 5.2, which confirms that the absolute errors are decreases with increasing order of approximations, and so, the approximate series solutions converge to the exact solutions. Moreover, the solutions converge fast for large Reynolds numbers* $(Re)$*. The behavior of the velocity field of classical Navier- Stokes equation via either method is depicted in Figure 5.1 while the behavior of the velocity field of time fractional Navier- Stokes equation with* $\alpha=0.1,0.5,0.8$ *is depicted in Figure 5.3.*

### 5.3.2 Solution of Example 5.3.2

**A. By fractional reduced differential transform method**

The following recurrence relation is obtained by taking FRDTM of Eq. (5.3.1)-(5.3.4),

$$\begin{cases} \dfrac{\Gamma(1+(1+k)\alpha)}{\Gamma(1+k\alpha)}U_\alpha^{k+1}(X)+\displaystyle\sum_{\ell=0}^{k}\left(\frac{\partial U_\alpha^{\ell}(X)}{\partial x}U_\alpha^{k-\ell}(X)+\frac{\partial U_\alpha^{\ell}(X)}{\partial y}V_\alpha^{k-\ell}(X)\right) \\ =\nu\nabla^2(U_\alpha^k(X))+g_1\delta(k), \quad U_\alpha^0(X)=-e^{x+y}, \\ \dfrac{\Gamma(1+(1+k)\alpha)}{\Gamma(1+k\alpha)}V_\alpha^{k+1}(X)+\displaystyle\sum_{\ell=0}^{k}\left(\frac{\partial V_\alpha^{\ell}(X)}{\partial x}U_\alpha^{k-\ell}(X)+\frac{\partial V_\alpha^{\ell}(X)}{\partial y}V_{k-\ell}(X)\right) \\ =\nu\nabla^2(V_\alpha^k(X))-g_1\delta(k), \quad V_\alpha^0(X)=e^{x+y},\ k\geq 0. \end{cases} \tag{5.3.16}$$

On solving the system (5.3.16), we have

$$\begin{aligned} U_\alpha^1(X)&=-\frac{2\nu}{\Gamma(1+\alpha)}e^{x+y}+\frac{g}{\Gamma(1+\alpha)}; & V_\alpha^1(X)&=\frac{2\nu}{\Gamma(1+\alpha)}e^{x+y}-\frac{g}{\Gamma(1+\alpha)} \\ U_\alpha^2(X)&=-\frac{(2\nu)^2}{\Gamma(1+2\alpha)}e^{x+y}; & V_\alpha^2(X)&=\frac{(2\nu)^2}{\Gamma(1+2\alpha)}e^{x+y} \\ U_\alpha^3(X)&=-\frac{(2\nu)^3}{\Gamma(1+3\alpha)}e^{x+y}; & V_\alpha^3(X)&=\frac{(2\nu)^3}{\Gamma(1+3\alpha)}e^{x+y} \\ \vdots \quad & \quad \vdots \quad & \vdots \quad & \quad \vdots \\ U_\alpha^k(X)&=-\frac{(2\nu)^k}{\Gamma(1+k\alpha)}e^{x+y}; & V_\alpha^k(X)&=\frac{(2\nu)^k}{\Gamma(1+k\alpha)}e^{x+y},\ k\geq 2, \end{aligned} \tag{5.3.17}$$

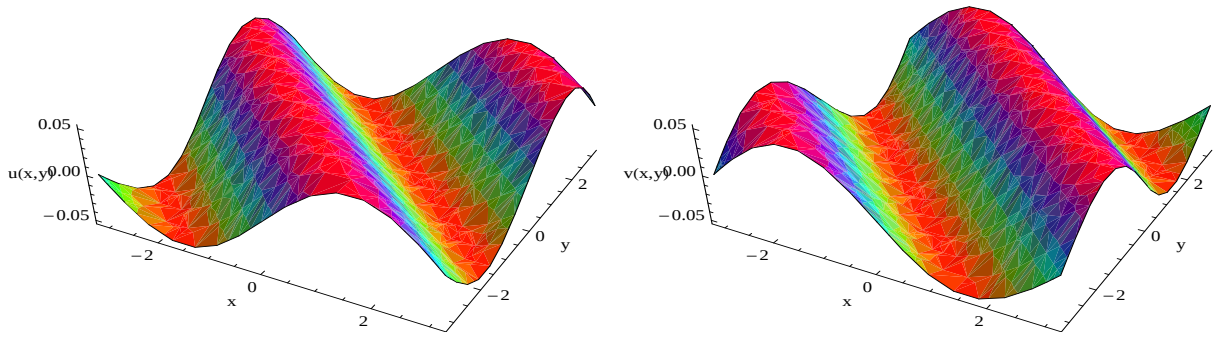


Figure 5.1: The behavior of $u$ and $v$ of Navier- Stokes equation in Example 5.3.1 at $t=3$ with the parameters $\alpha=1,\ g=0,\ \nu=0.5$

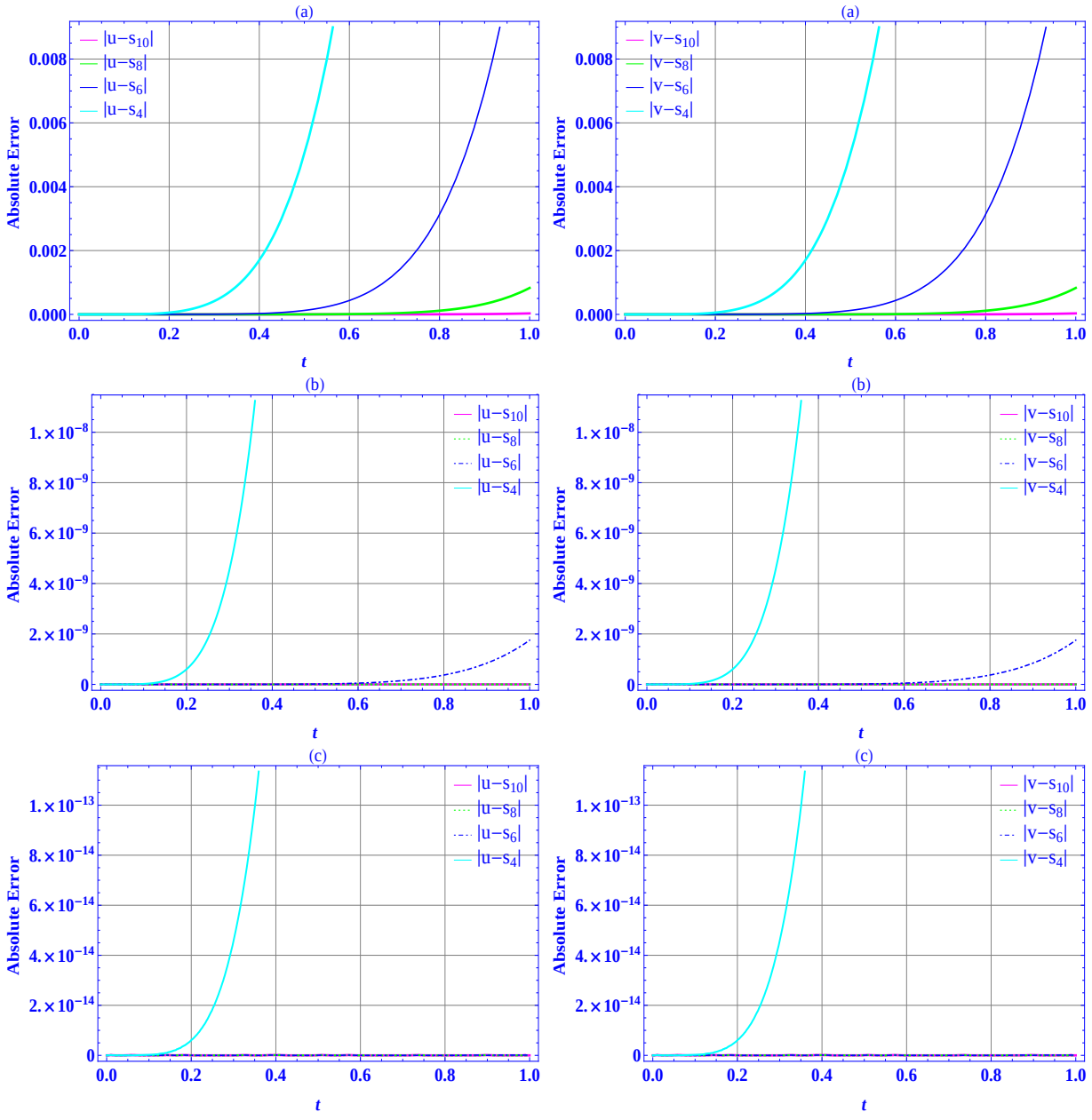


Figure 5.2: Absolute errors in different order ($m = 4, 6, 8, 10$) approximate solutions $u, v$ of Example 5.3.1 with different value of $Re$ a) $Re = 1$, b) $Re = 10$, c) $Re = 100$

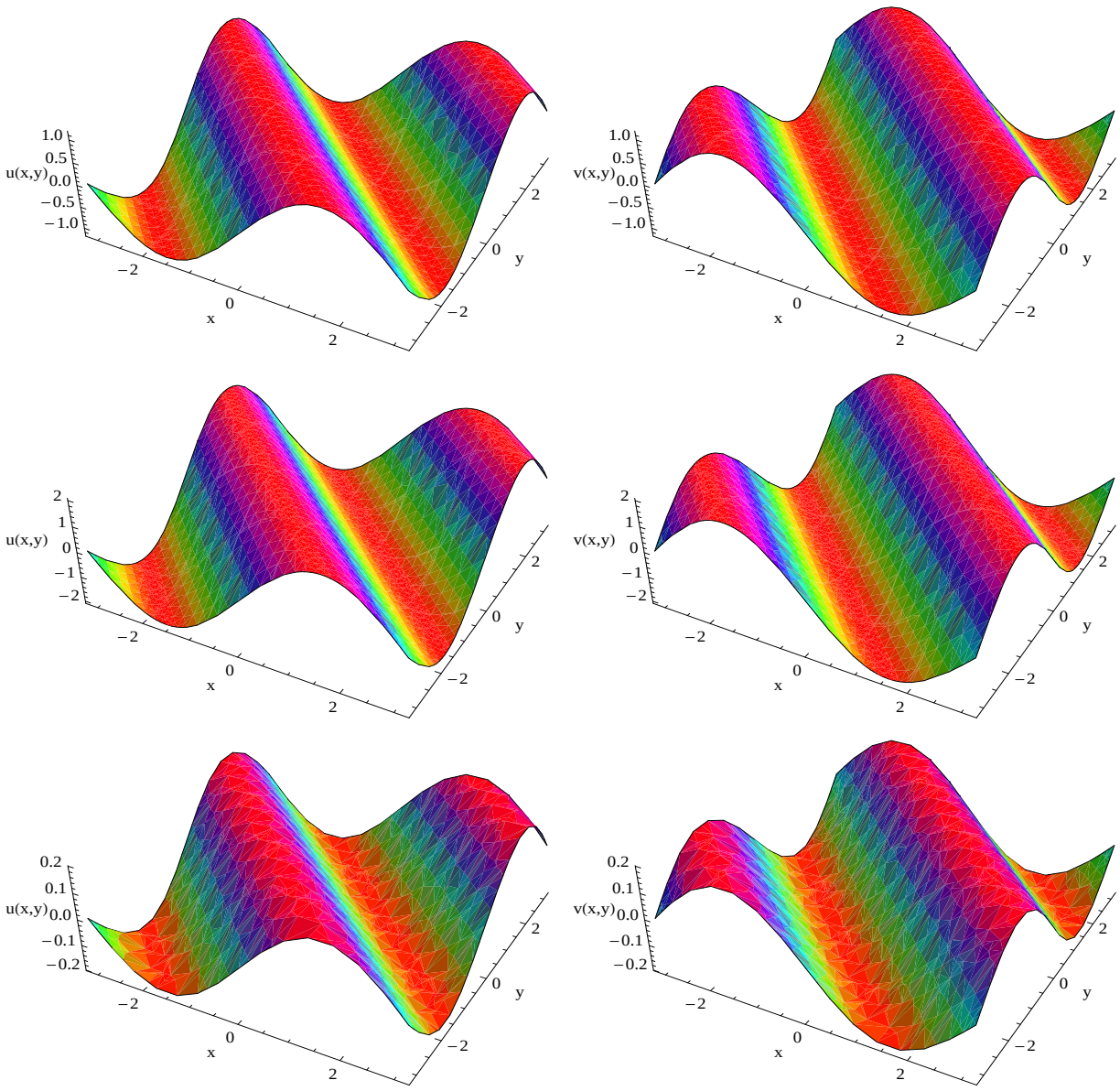


Figure 5.3: The behavior of $u$ and $v$ of Navier- Stokes equation in Example 5.3.1 at $t = 3$ with the parameters: $\alpha = 0.5,\ g = 0,\ \nu = 0.5$ (upper), $\alpha = 0.1,\ g = 0,\ \nu = 0.5$ (middle) and $\alpha = 0.8,\ g = 0,\ \nu = 0.5$ (lower)

By using inverse FRDT, we have

$$\begin{aligned} u(x,y,t) &= \sum_{k=0}^{\infty} U_\alpha^k(X) t^{\alpha k} = -e^{x+y} \sum_{k=0}^{\infty} \frac{(2\nu t^\alpha)^k}{\Gamma(1+k\alpha)} + \frac{g t^\alpha}{\Gamma(1+\alpha)}, \\ &= -e^{x+y} E_{\alpha,1}(2\nu t^\alpha) + \frac{g t^\alpha}{\Gamma(1+\alpha)}. \\ v(x,y,t) &= \sum_{k=0}^{\infty} V_\alpha^k(X) t^{\alpha k} = e^{x+y} \sum_{k=0}^{\infty} \frac{(2\nu t^\alpha)^k}{\Gamma(1+k\alpha)} - \frac{g t^\alpha}{\Gamma(1+\alpha)}, \\ &= e^{x+y} E_{\alpha,1}(2\nu t^\alpha) - \frac{g t^\alpha}{\Gamma(1+\alpha)}. \end{aligned} \tag{5.3.18}$$

which is the required approximate solution.

**B. By new integral projected differential transform method**

NIPDT of Navier- Stokes equation (5.3.1) with initial conditions (5.3.4) yields the following recurrence relation

$$\begin{aligned} U(X,h+1) &= \mathcal{K}^{-1}\left(\mu^{2\alpha}\mathcal{K}\left[A_h^{(1)}+B_h^{(1)}\right]\right), u(X,0) = -e^{x+y} + \frac{gt^\alpha}{\Gamma(1+\alpha)} \\ V(X,h+1) &= \mathcal{K}^{-1}\left(\mu^{2\alpha}\mathcal{K}\left[A_h^{(2)}+B_h^{(2)}\right]\right), u(X,0) = e^{x+y} - \frac{gt^\alpha}{\Gamma(1+\alpha)} \end{aligned} \tag{5.3.19}$$

where $A_h^{(\ell)}, B_h^{(\ell)}$ $(\ell = 1, 2)$ are same as defined in Eq. (5.3.12). The relation (5.3.19) yields

$$\begin{aligned} &A_0^{(1)} = 0, && B_0^{(1)} = -2\nu e^{x+y}, && U(X,1) = -\tfrac{2\nu t^\alpha}{\Gamma(\alpha+1)}e^{x+y}, \\ &A_0^{(2)} = 0, && B_0^{(2)} = 2\nu e^{x+y}, && V(X,1) = \tfrac{2\nu t^\alpha}{\Gamma(\alpha+1)}e^{x+y}, \\ &A_1^{(1)} = 0, && B_1^{(1)} = -\tfrac{(2\nu)^2 t^\alpha}{\Gamma(\alpha+1)}e^{x+y}, && U(X,2) = -\tfrac{(2\nu t^\alpha)^2}{\Gamma(2\alpha+1)}e^{x+y}, \\ &A_1^{(2)} = 0, && B_1^{(2)} = \tfrac{(2\nu)^2 t^\alpha}{\Gamma(\alpha+1)}e^{x+y}, && V(X,2) = \tfrac{(2\nu t^\alpha)^2}{\Gamma(2\alpha+1)}e^{x+y}, \\ &A_2^{(1)} = 0, && B_2^{(1)} = -\tfrac{(2\nu)^3 t^{2\alpha}}{\Gamma(2\alpha+1)}e^{x+y}, && U(X,3) = -\tfrac{(2\nu t^\alpha)^3}{\Gamma(3\alpha+1)}e^{x+y}, \\ &A_2^{(2)} = 0, && B_2^{(2)} = \tfrac{(2\nu)^3 t^{2\alpha}}{\Gamma(2\alpha+1)}e^{x+y}, && V(X,3) = \tfrac{(2\nu t^\alpha)^3}{\Gamma(3\alpha+1)}e^{x+y}, \\ &\vdots && \vdots && \vdots \end{aligned} \tag{5.3.20}$$

Hence, the solution $u(X,t), v(X,t)$ are obtained as

$$\begin{aligned} u(X,t) &= \sum_{h=0}^{\infty} u(X,h) = -e^{x+y}\sum_{h=0}^{\infty}\frac{(2\nu t^\alpha)^h}{\Gamma(1+h\alpha)} + \frac{gt^\alpha}{\Gamma(1+\alpha)} \\ &= -e^{x+y}E_{\alpha,1}(2\nu t^\alpha) + \frac{gt^\alpha}{\Gamma(1+\alpha)}, \\ v(X,t) &= \sum_{h=0}^{\infty} v(X,h) = e^{x+y}\sum_{h=0}^{\infty}\frac{(2\nu t^\alpha)^h}{\Gamma(1+h\alpha)} - \frac{gt^\alpha}{\Gamma(1+\alpha)} \\ &= e^{x+y}E_{\alpha,1}(2\nu t^\alpha) - \frac{gt^\alpha}{\Gamma(1+\alpha)}. \end{aligned} \tag{5.3.21}$$

**Remark 5.3.5.** *It is noticed that solutions* (5.3.18) *and* (5.3.21) *are same. The same solutions for* $g = 0$ *are obtained by using discrete adomian decomposition method [34]. In particular the above solution with* $g = 0$ *and* $\alpha = 1$ *is the closed form of the exact solution* (5.3.5) *of the classical Navier- Stokes equation. The absolute errors in different*

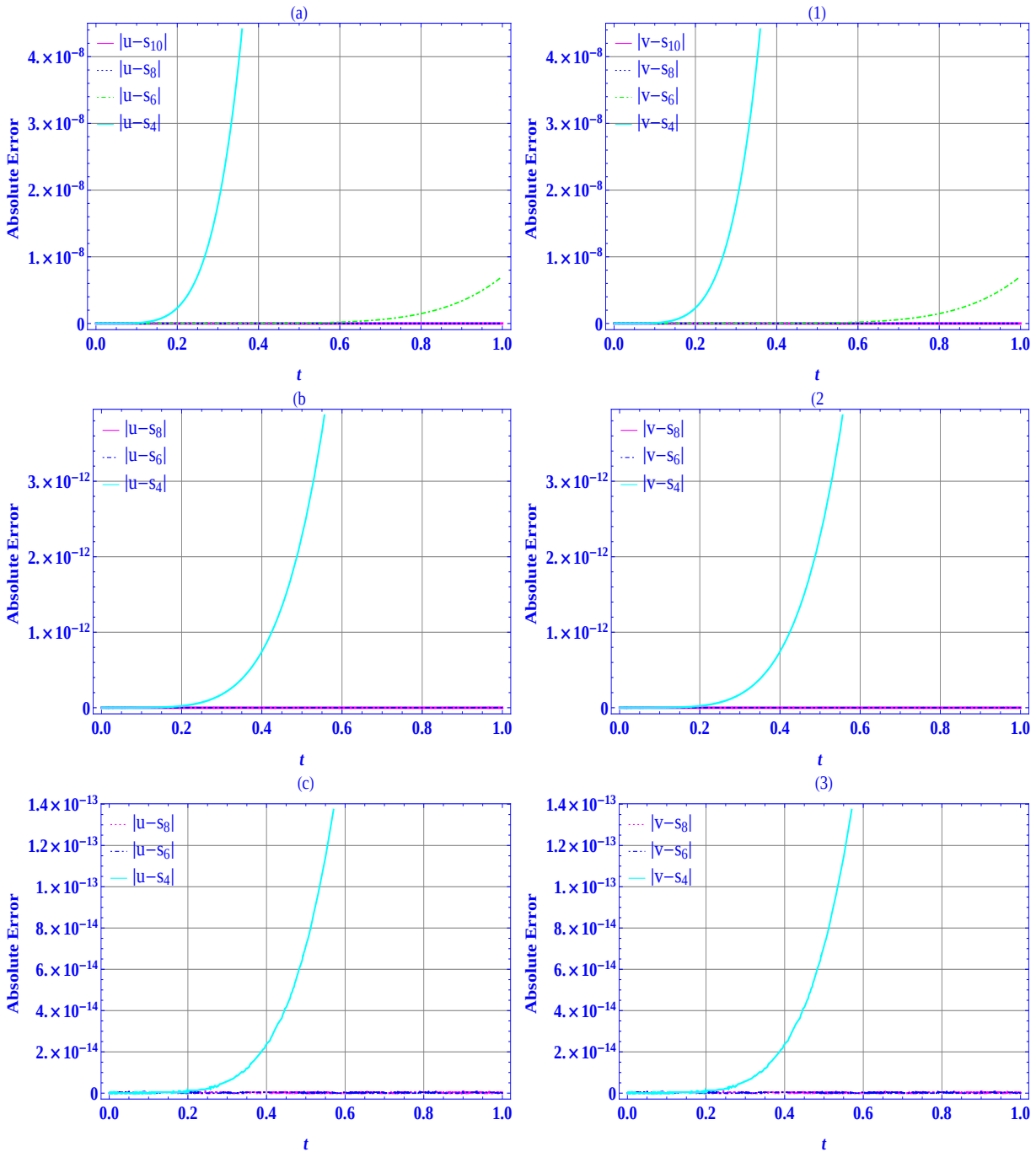


Figure 5.4: Absolute errors in different order $(m = 4, 6, 8, 10)$ approximate solutions $u, v$ of Example 5.3.2 with different value of $Re$ a) $Re = 10$, b) $Re = 100$, c) $Re = 500$

*order* $(m = 4, 6, 8, 10)$ *approximate solutions of velocity profile* $(u, v)$ *of Example 5.3.2 with the different value of* $Re = 10, 100, 500$ *is depicted in Figure 5.4, which confirms that the absolute errors are decreases with increasing order of approximations, and so, the approximate series solutions converge to the exact solutions. Moreover, the solutions converge fast for large Reynolds numbers* $(Re)$*. The behavior of velocity field of classical Navier- Stokes equation via either method is depicted for* $\alpha = 1, 0.5$ *in Figure 5.5.*

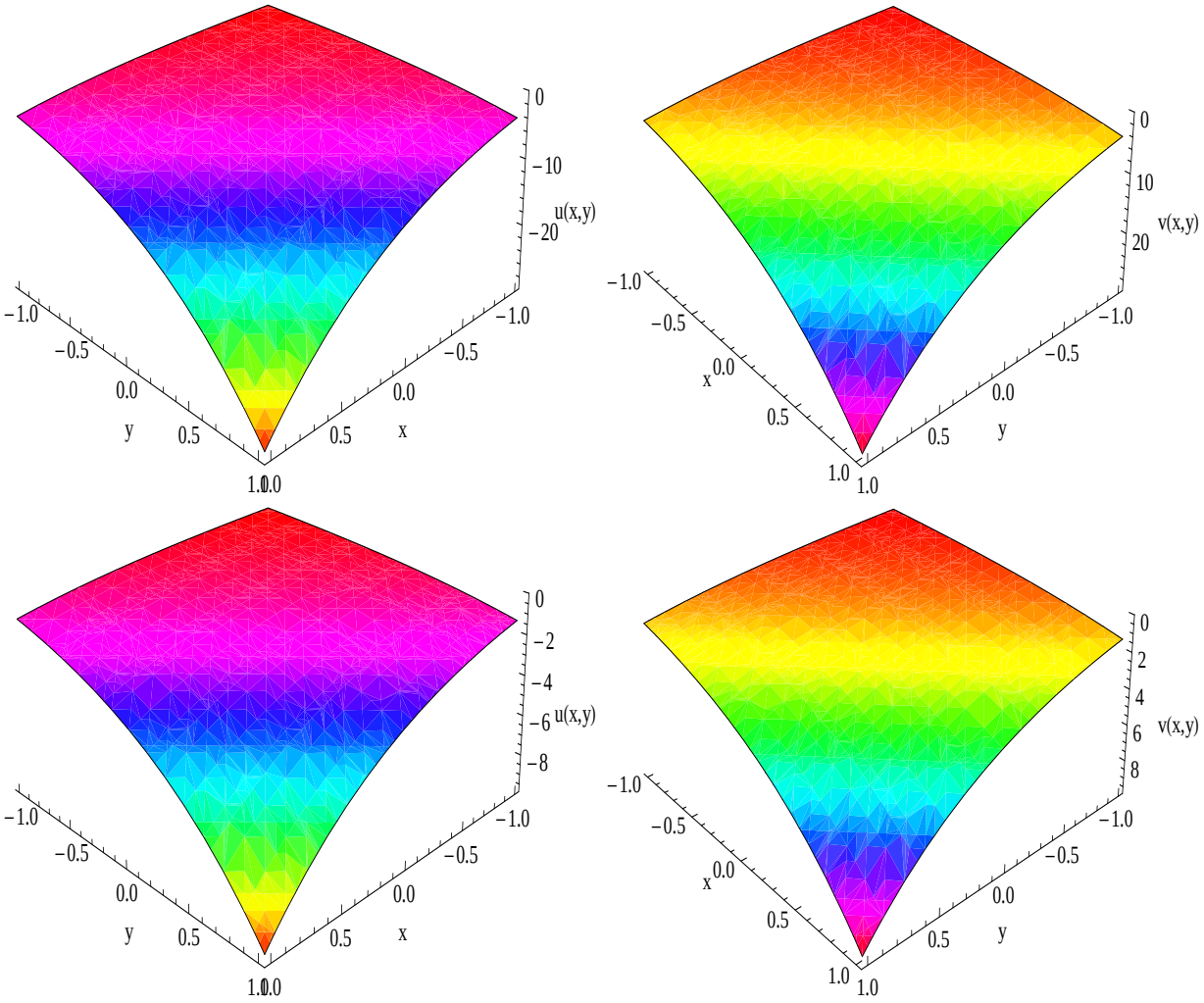


Figure 5.5: The behavior of $u$ and $v$ of Navier- Stokes equation in Example 5.3.2 with the parameters $g = 0,\ \nu = 0.5$ and $\alpha = 1$(upper), $\alpha = 0.5$ at $t = 0.05$ (lower)

### 5.3.3 Solution of Example 5.3.3

**A. By reduced differential transform method**

The FRDT of time fractional Navier-Stoke's equation considered in Example 5.3.3 yields the following recurrence relation

$$\begin{cases} \dfrac{\Gamma(1+(1+k)\alpha)}{\Gamma(1+k\alpha)}U_\alpha^{k+1}(X) = \nu\nabla^2(U_\alpha^k(X)) \\ -\displaystyle\sum_{\ell=0}^{k}\left(\frac{\partial U_\alpha^\ell(X)}{\partial x}U_\alpha^{k-\ell}(X) + \frac{\partial U_\alpha^\ell(X)}{\partial y}V_\alpha^{k-\ell}(X) + \frac{\partial U_\alpha^\ell(X)}{\partial z}W_\alpha^{k-\ell}(X)\right), \\ \dfrac{\Gamma(1+(1+k)\alpha)}{\Gamma(1+k\alpha)}V_\alpha^{k+1}(X) = \nu\nabla^2(V_\alpha^k(X)) \\ -\displaystyle\sum_{\ell=0}^{k}\left(\frac{\partial V_\alpha^\ell(X)}{\partial x}U_\alpha^{k-\ell}(X) + \frac{\partial V_\alpha^\ell(X)}{\partial y}V_\alpha^{k-\ell}(X) + \frac{\partial V_\alpha^\ell(X)}{\partial z}W_\alpha^{k-\ell}(X)\right), \\ \dfrac{\Gamma(1+(1+k)\alpha)}{\Gamma(1+k\alpha)}W_\alpha^{k+1}(X) = \nu\nabla^2(W_\alpha^k(X)) \\ -\displaystyle\sum_{\ell=0}^{k}\left(\frac{\partial W_\alpha^\ell(X)}{\partial x}U_\alpha^{k-\ell}(X) + \frac{\partial W_\alpha^\ell(X)}{\partial y}V_\alpha^{k-\ell}(X) + \frac{\partial W_\alpha^\ell(X)}{\partial z}W_\alpha^{k-\ell}(X)\right), \\ U_\alpha^0(X) = -0.5x + y + z,\ V_\alpha^0(X) = x - 0.5y + z,\ W_\alpha^0(X) = x + y - 0.5z. \end{cases} \tag{5.3.22}$$

On solving the simultaneous equations in (5.3.22), we get

$$U_\alpha^1(X) = -\frac{2.25}{\Gamma(1+\alpha)}x; U_\alpha^2(X) = \frac{2(2.25)}{\Gamma(1+2\alpha)}U_\alpha^0(X); U_\alpha^3(X) = -\frac{(2.25)^2}{\Gamma(1+3\alpha)}\left(4 + \frac{\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2}\right)x;$$
$$U_\alpha^4(X) = \frac{(2.25)^2}{\Gamma(1+4\alpha)}\left(8 + \frac{2\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2} + \frac{4\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)}\right)U_\alpha^0(X); \ldots$$

$$V_\alpha^1(X) = -\frac{2.25}{\Gamma(1+\alpha)}x; V_\alpha^2(X) = \frac{2(2.25)}{\Gamma(1+2\alpha)}V_\alpha^0(X); V_\alpha^3(X) = -\frac{(2.25)^2}{\Gamma(1+3\alpha)}\left(4 + \frac{\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2}\right)y;$$
$$V_\alpha^4(X) = \frac{(2.25)^2}{\Gamma(1+4\alpha)}\left(8 + \frac{2\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2} + \frac{4\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)}\right)V_\alpha^0(X); \ldots$$

$$W_\alpha^1(X) = -\frac{2.25}{\Gamma(1+\alpha)}z; W_\alpha^2(X) = \frac{2(2.25)}{\Gamma(1+2\alpha)}W_\alpha^0(X); W_\alpha^3(X) = -\frac{(2.25)^2}{\Gamma(1+3\alpha)}\left(4 + \frac{\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2}\right)z;$$
$$W_\alpha^4(X) = \frac{(2.25)^2}{\Gamma(1+4\alpha)}\left(8 + \frac{2\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2} + \frac{4\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)}\right)W_\alpha^0(X); \ldots$$

By using inverse FRDT, we have

$$
\begin{aligned}
u(x,y,z,t) &= \sum_{k=0}^{\infty} U_\alpha^k(X) = -0.5x + y + z - \frac{2.25}{\Gamma(1+\alpha)} x t^\alpha + \frac{2(2.25)}{\Gamma(1+2\alpha)}(-0.5x + y + z)t^{2\alpha} \\
&\quad - \frac{(2.25)^2}{\Gamma(1+3\alpha)}\left(4 + \frac{\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2}\right) x t^{3\alpha} \\
&\quad + \frac{(2.25)^2}{\Gamma(1+4\alpha)}\left(8 + \frac{2\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2} + \frac{4\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)}\right)(-0.5x + y + z)t^{4\alpha} + \ldots \\
v(x,y,z,t) &= \sum_{k=0}^{\infty} V_\alpha^k(X) = x - 0.5y + z - \frac{2.25}{\Gamma(1+\alpha)} y t^\alpha + \frac{2(2.25)}{\Gamma(1+2\alpha)}(x - 0.5y + z)t^{2\alpha} \\
&\quad - \frac{(2.25)^2}{\Gamma(1+3\alpha)}\left(4 + \frac{\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2}\right) y t^{3\alpha} \\
&\quad + \frac{(2.25)^2}{\Gamma(1+4\alpha)}\left(8 + \frac{2\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2} + \frac{4\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)}\right)(x - 0.5y + z)t^{4\alpha} + \ldots \\
w(x,y,z,t) &= \sum_{k=0}^{\infty} W_\alpha^k(X) = x + y - 0.5z - \frac{2.25}{\Gamma(1+\alpha)} z t^\alpha + \frac{2(2.25)}{\Gamma(1+2\alpha)}(x + y - 0.5z)t^{2\alpha} \\
&\quad - \frac{(2.25)^2}{\Gamma(1+3\alpha)}\left(4 + \frac{\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2}\right) z t^{3\alpha} \\
&\quad + \frac{(2.25)^2}{\Gamma(1+4\alpha)}\left(8 + \frac{2\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2} + \frac{4\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)}\right)(x + y - 0.5z)t^{4\alpha} + \ldots
\end{aligned}
$$

which is the required exact solution. In particular for $\alpha = 1$, we get

$$
\left.
\begin{aligned}
u(X,t) &= \left(-\frac{x}{2} + y + z\right)\left(1 + \frac{9}{4}t^2 + \left(\frac{9}{4}\right)^2 t^4 + \ldots\right) - \frac{9}{4}xt\left(1 + \frac{9}{4}t^2 + \ldots\right) \\
&= \frac{-\frac{x}{2} + y + z - \frac{9xt}{4}}{1 - \frac{9t^2}{4}}. \\
v(X,t) &= \left(x - \frac{y}{2} + z\right)\left(1 + \frac{9}{4}t^2 + \left(\frac{9}{4}\right)^2 t^4 + \ldots\right) - \frac{9}{4}yt\left(1 + \frac{9}{4}t^2 + \ldots\right) \\
&= \frac{x - \frac{y}{2} + z - \frac{9yt}{4}}{1 - \frac{9t^2}{4}}. \\
w(X,t) &= \left(x + y - \frac{z}{2}\right)\left(1 + \frac{9}{4}t^2 + \left(\frac{9}{4}\right)^2 t^4 + \ldots\right) - \frac{9}{4}zt\left(1 + \frac{9}{4}t^2 + \ldots\right) \\
&= \frac{x + y - \frac{z}{2} - \frac{9zt}{4}}{1 - \frac{9t^2}{4}}.
\end{aligned}
\right\} \quad (5.3.23)
$$

which is the closed form of the exact solution of the associated classical Navier- Stokes equation.

**B. By new integral projected differential transform method**

On applying NIPDT in (5.1.2) with (5.3.6) yields the following recurrence relation

$$\begin{aligned} U(X,h+1) &= \mathcal{K}^{-1}\left(\mu^{2\alpha}\mathcal{K}\left[A_h^{(1)}+B_h^{(1)}\right]\right), U(X,0) = -\frac{x}{2}+y+z \\ V(X,h+1) &= \mathcal{K}^{-1}\left(\mu^{2\alpha}\mathcal{K}\left[A_h^{(2)}+B_h^{(2)}\right]\right), V(X,0) = x-\frac{y}{2}+z \\ W(X,h+1) &= \mathcal{K}^{-1}\left(\mu^{2\alpha}\mathcal{K}\left[A_h^{(3)}+B_h^{(3)}\right]\right), W(X,0) = x+y-\frac{z}{2} \end{aligned} \tag{5.3.24}$$

where $A_h^{(\ell)}, B_h^{(\ell)},\ (\ell = 1,2,3)$ are defined in Eq.(5.2.10). On solving the above relation, we get $B_h^{(\ell)} = 0$ for each $\ell = 1,2,3$ and $h = 0,1,2,\ldots$, and so,

$$\begin{aligned} A_0^{(1)} &= -\tfrac{9x}{4}, \quad U(X,1) = -\tfrac{9xt^{\alpha}}{4\Gamma(\alpha+1)}, \\ A_0^{(2)} &= -\tfrac{9y}{4}, \quad V(X,1) = -\tfrac{9yt^{\alpha}}{4\Gamma(\alpha+1)}, \\ A_0^{(3)} &= -\tfrac{9z}{4}, \quad W(X,1) = -\tfrac{9zt^{\alpha}}{4\Gamma(\alpha+1)}, \end{aligned}$$

$$\begin{aligned} A_1^{(1)} &= -\tfrac{9U(X,0)t^{\alpha}}{2\Gamma(\alpha+1)}, & U(X,2) &= \tfrac{9U(X,0)t^{2\alpha}}{2\Gamma(2\alpha+1)}, \\ A_1^{(2)} &= -\tfrac{9V(X,0)t^{\alpha}}{2\Gamma(\alpha+1)}, & V(X,2) &= \tfrac{9V(X,0)t^{2\alpha}}{2\Gamma(2\alpha+1)}, \\ A_1^{(3)} &= -\tfrac{9W(X,0)t^{\alpha}}{2\Gamma(\alpha+1)}, & W(X,2) &= \tfrac{9W(X,0)t^{2\alpha}}{2\Gamma(2\alpha+1)}, \\ A_2^{(1)} &= -\left(4+\tfrac{\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}\right)\tfrac{81xt^{2\alpha}}{16\Gamma(1+2\alpha)}, & U(X,3) &= \tfrac{-81xt^{3\alpha}}{16\Gamma(1+3\alpha)}\left(4+\tfrac{\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}\right), \\ A_2^{(2)} &= -\left(4+\tfrac{\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}\right)\tfrac{81yt^{2\alpha}}{16\Gamma(1+2\alpha)}, & V(X,3) &= \tfrac{-81yt^{3\alpha}}{16\Gamma(1+3\alpha)}\left(4+\tfrac{\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}\right), \\ A_2^{(3)} &= -\left(4+\tfrac{\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}\right)\tfrac{81zt^{2\alpha}}{16\Gamma(1+2\alpha)}, & W(X,3) &= \tfrac{-81zt^{3\alpha}}{16\Gamma(1+3\alpha)}\left(4+\tfrac{\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}\right), \end{aligned}$$

$$\begin{aligned} A_3^{(1)} &= \left(8+\tfrac{4\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)}+\tfrac{2\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}\right)\tfrac{81U(X,0)t^{3\alpha}}{16\Gamma(1+3\alpha)}, \\ U(X,4) &= \left(8+\tfrac{4\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)}+\tfrac{2\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}\right)\tfrac{81U(X,0)t^{4\alpha}}{16\Gamma(1+4\alpha)}, \\ A_3^{(2)} &= \left(8+\tfrac{4\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)}+\tfrac{2\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}\right)\tfrac{81V(X,0)t^{3\alpha}}{16\Gamma(1+3\alpha)}, \\ V(X,4) &= \left(8+\tfrac{4\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)}+\tfrac{2\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}\right)\tfrac{81V(X,0)t^{4\alpha}}{16\Gamma(1+4\alpha)}, \end{aligned}$$

$$\begin{aligned} A_3^{(3)} &= \left(8+\tfrac{4\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)}+\tfrac{2\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}\right)\tfrac{81W(X,0)t^{3\alpha}}{16\Gamma(1+3\alpha)}, \\ W(X,4) &= \left(8+\tfrac{4\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)}+\tfrac{2\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}\right)\tfrac{81W(X,0)t^{4\alpha}}{16\Gamma(1+4\alpha)}, \vdots \end{aligned}$$

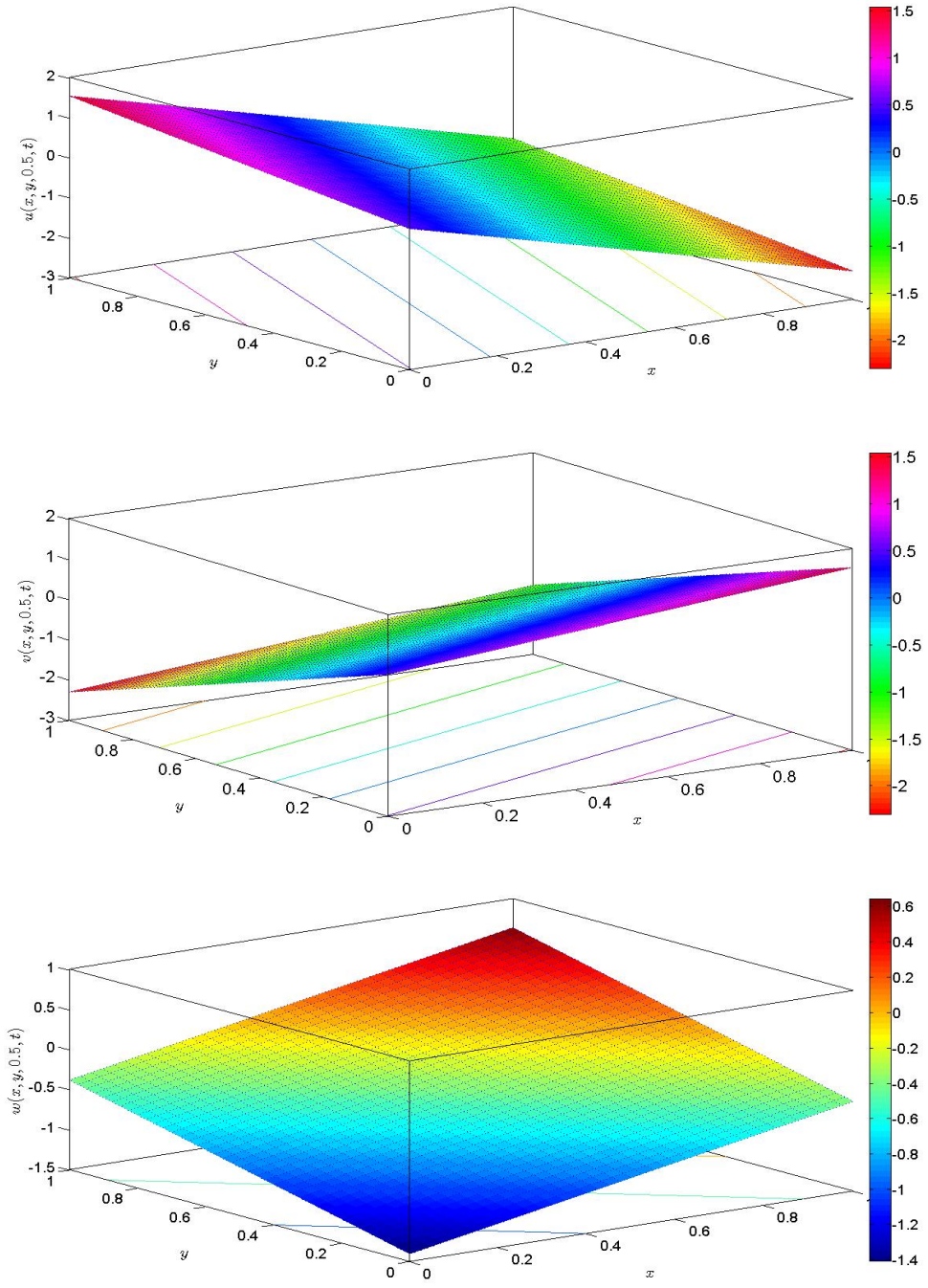


Figure 5.6: The velocity profile $(u, v, w)$ of Navier- Stokes equation in Example 5.3.3 at $t = 0.1$ with $\alpha = 1$

Thus, the solution is given by

$$u(X,t) = U(X,0) + U(X,1) + U(X,2) + U(X,3) + \dots$$
$$= -\frac{x}{2} + y + z - \frac{9x}{4\Gamma(1+\alpha)}t^{\alpha} + \frac{9t^{2\alpha}}{2\Gamma(1+2\alpha)}\left(-\frac{x}{2} + y + z\right) - \left(4 + \frac{\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}\right) \times$$
$$\frac{81xt^{3\alpha}}{16\Gamma(1+3\alpha)} + \frac{81t^{4\alpha}}{16\Gamma(1+4\alpha)}\left(8 + \frac{4\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)} + \frac{2\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}\right)\left(-\frac{x}{2} + y + z\right) + \dots$$

$$v(X,t) = V(X,0) + V(X,1) + V(X,2) + V(X,3) + \dots$$
$$= x - \frac{y}{2} + z - \frac{9y}{4\Gamma(1+\alpha)}t^{\alpha} + \frac{9t^{2\alpha}}{2\Gamma(1+2\alpha)}\left(x - \frac{y}{2} + z\right) - \left(4 + \frac{\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}\right)\frac{81yt^{3\alpha}}{16\Gamma(1+3\alpha)}$$
$$+ \frac{81t^{4\alpha}}{16\Gamma(1+4\alpha)}\left(8 + \frac{4\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)} + \frac{2\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}\right)\left(x - \frac{y}{2} + z\right) + \dots$$

$$w(X,t) = W(X,0) + W(X,1) + W(X,2) + W(X,3) + \dots$$
$$= x + y - \frac{z}{2} - \frac{9z}{4\Gamma(1+\alpha)}t^{\alpha} + \frac{9t^{2\alpha}}{2\Gamma(1+2\alpha)}\left(x + y - \frac{z}{2}\right) - \left(4 + \frac{\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}\right)\frac{81zt^{3\alpha}}{16\Gamma(1+3\alpha)}$$
$$+ \frac{81t^{4\alpha}}{16\Gamma(1+4\alpha)}\left(8 + \frac{4\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)} + \frac{2\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}\right)\left(x + y - \frac{z}{2}\right) + \dots$$

**Remark 5.3.6.** *The same solution is obtained FRDTM and NIPDTM are same. The proposed solutions for* $\alpha = 1$ *by either of these method reduces to the solutions* (5.3.23)*. The same solutions are obtained by using high-order finite difference scheme with a linearization technique [38]. The velocity profile* $(u, v, w)$ *of the Navier-Stokes equation for* $\alpha = 1$ *is depicted in Figure 5.6, which shows the same characteristics as reported in [38].*

## 5.4 Conclusion

In this chapter two reliable methods: `fractional reduced differential transform method` and `new integral projected differential transform method` have been adopted for the numerical simulation of time-fractional model of the Navier-Stokes equations with suitable initial conditions. The fractional derivative is considered in the Caputo sense. The analytical results have been given in terms of a power series. Three test problems are carried out in order to validate and illustrate the efficiency of the method. The findings are as follows

- ✓ The proposed solutions from both of the methods are identical.
- ✓ The proposed solutions agree excellently with HPM [68], a high-order finite difference scheme with a linearization technique [38] and ADM [34].
- ✓ The approximate series solutions converge to the exact solutions. Moreover, the solutions converge fast for large Reynolds numbers ($Re$).
- ✓ The FRDTM solutions are approximated without any discretization, transformation, perturbation, or restrictive conditions. However, the performed calculations in FRDTM show that the described it needs a very small size of computation in comparison to HPM [68], ADM [34], high-order finite difference scheme with a linearization technique [38] and NIPDTM.
- ✓ Small size of computation contrary to the other schemes, is the strength of FRDTM scheme.

# Chapter 6

# An approximate solution of time fractional coupled viscous Burgers' equation in multi-dimensions via homotopy perturbation method

## 6.1 Introduction

The study of the solution behavior of fractional partial differential is very popular area of the research as it has a great attentions among the researchers due to its applications in modeling of various nonlinear complex systems arising in fluid mechanics, viscoelasticity, mathematical biology, life sciences, electrochemistry and physics [22, 39, 40, 166, 194]. Most of the cases, it is very tough to compute the exact behavior of the fractional differential equations. A great deal of effort has been expanded to develop techniques for computations of the approximate behavior of such type of equations. Especially, in nano-hydrodynamics where continuum assumption does not well, the fractional model can be considered to be the best candidate. In the past years, several vigorous techniques have been proposed for solving such type of fractional PDEs, among them, homotopy perturbation method (HPM) [86], homotopy perturbation Sumudu transform method [243], homotopy analysis

method [200], adomian decomposition method (ADM) [291], reduced differential transform method [204, 211], fractional reduced differential transform method (FRDTM) [246, 248, 252], variation iteration method [193].

The computation of the solutions time fractional coupled Burgers' equation (6.3.1) is of great importance due to its applications in approximate theory of flow through a shock wave traveling in a viscous fluid [52], in model of turbulence [36]. Burgers' equation is reduced to a simple model of sedimentation or evolution of scaled volume concentration of two/three kinds of particles in fluid suspension or colloids under the effect of gravity modulation [182]. The fractional model of one dimensional coupled viscous Burgers' equation has been solved by employing reduced differential transform [247, 251, 256], fractional reduced differential transform [237], generalized differential transform method [155], coupling of HPM and Padè technique [108], HPM [85], ADM [291], Laplace homotopy algorithm [243], Laplace homotopy perturbation method [103] and Fractional variational iteration method [193]. For more details of HPM, we refer the readers to [20, 187, 206, 286] and references therein.

This chapter deals with an approximate behavior of multi-dimensional, time fractional coupled viscous Burgers' equation obtained by employing He's HPM [85].

### 6.1.1 Homotopy perturbation method

Consider the boundary value differential equation of the form

$$L(u) + N(u) - f(r) = 0, \quad r \in \Omega, \tag{6.1.1}$$

$$\mathcal{B}\left(u, \frac{\partial u}{\partial n}\right), \quad r \in \partial\Omega \tag{6.1.2}$$

where $f(r)$ be a known analytical function, $\mathcal{B}$ be the operator for boundary conditions, $\partial\Omega$ be the boundary of the domain $\Omega$ whereas $L$ and $N$ represent linear and nonlinear differential operators, respectively.

Define homotopy $\nu(r, p) : \Omega \times [0, 1] \to \mathcal{R}$ for (6.1.1)-(6.1.2) as in [86] satisfying

$$\mathcal{H}(\nu, p) = (1 - p)\left[L(\nu) - L(u_0)\right] + p\left[L(\nu) + N(\nu) - f(r)\right] = 0, \tag{6.1.3}$$

where $p \in [0,1]$ is an embedding parameter, $u_0$ is an initial approximation obeying condition (6.1.2). It is worth mentioning that Eq. (6.1.3) reduces to linearized differential equation: $L(\nu) = L(u_0)$ for $p = 0$ while nonlinear original differential equation (6.1.1) for $p = 1$. Thus, the changing process of $p$ from zero to unity in (6.1.3) results the corresponding changes in $\nu(r,p)$ from $u_0$ to $u(r)$. In topology, this process is referred to as deformation. The equations $L(\nu) - L(u_0)$ and $L(\nu) + N(\nu) - f(r)$ are called homotopic equations.

The basis solution of (6.1.3) is taken as

$$\nu = \nu_0 + p\nu_1 + p^2\nu_2 + \dots \tag{6.1.4}$$

Moreover, solution (6.1.4) converges to $u$ (i.e., $\nu \to u$) as $p \to 1$ [33], and so,

$$u = \nu_0 + \nu_1 + \nu_2 + \dots \tag{6.1.5}$$

### 6.1.2 HPM for time-fractional nonlinear PDEs

Consider initial value system of time-fractional nonlinear differential equation as follows:

$$\mathcal{D}_t^{\alpha} u(X,t) + L[u(X,t)] + N[u(X,t)] = f(X,t), \quad t > 0, \quad m-1 < \alpha < m, \tag{6.1.6}$$

$$u_0^{(k)}(X) = c_k(X), \qquad k = 0,1,2,\dots,m-1. \tag{6.1.7}$$

where $X = (x,y,z) \in \Omega \subset \mathbb{R}^3$, $L$ and $N$ denote linear and nonlinear differential operators, respectively. Either of these operator may include other fractional derivatives of order less than $\alpha$, $f \to$ known smooth function and $D_t^{\alpha} \to$Caputo fractional derivative of order $\alpha$.

Define homotopy for (6.1.6) as in [178] satisfying the following

$$\begin{aligned} &\frac{\partial^m u(X,t)}{\partial t^m} - f(X,t) \\ &\quad = p\left[\frac{\partial^m u(X,t)}{\partial t^m} - L[u(X,t)] - N[u(X,t)] - D_t^{\alpha} u(X,t)\right], \ p \in [0,1]. \end{aligned} \tag{6.1.8}$$

Moreover, for $p = 0$ Eq. (6.1.8) reduces to the linearized equation $u^{(m)}(X,t) = f(X,t)$.

Let the basic solution of (6.1.8) is considered as

$$u(X,t) = \sum_{\ell=0}^{\infty} p^{\ell} u_{\ell}, \tag{6.1.9}$$

and so, nonlinear term $N[u(X,t)]$ is decomposed as follows

$$N[u(X,t)] = \sum_{\ell=0}^{\infty} p^{\ell} H_{\ell}(u), \quad H_{\ell}(u) = \frac{1}{\ell!} \left\{ \frac{\partial^{\ell}}{\partial p^{\ell}} N \left[ \sum_{i=0}^{\infty} p^{i} u_i(X,t) \right] \right\}_{p=0} \tag{6.1.10}$$

where $H_{\ell}(u) \equiv H_{\ell}\left(u_0(X,t), u_1(X,t), \ldots, u_{\ell}(X,t)\right)$ is referred to as He's polynomial [86].

After implementing Eq. (6.1.9) - (8.2.6), Eq. (6.1.8) reduces to

$$\begin{aligned} &\sum_{\ell=0}^{\infty} p^{\ell} \frac{\partial^m u_{\ell}(X,t)}{\partial t^m} - f(X,t) \\ &\qquad = p \sum_{\ell=0}^{\infty} p^{\ell} \left[ \frac{\partial^m u_{\ell}(X,t)}{\partial t^m} - L[u_{\ell}(X,t)] - \mathcal{H}_{\ell}(u) - D_t^{\alpha} u_{\ell}(X,t) \right]. \end{aligned} \tag{6.1.11}$$

On equating the coefficient of like power $p^{\ell}$ $(\ell = 0, 1, 2, \ldots)$ on both sides of (6.1.11), a sequence of linear differential equation for each $u_{\ell}(X,t)$ is obtained as follows:

$$\begin{cases} \text{Coefficient of } p^0 : \dfrac{\partial^m u_0(X,t)}{\partial t^m} = f(X,t), \\ \text{Coefficient of } p^{\ell+1} : \dfrac{\partial^m u_{\ell+1}(X,t)}{\partial t^m} = \dfrac{\partial^m u_{\ell}(X,t)}{\partial t^m} - \mathcal{L}[u_{\ell}(X,t)] - \mathcal{H}_{\ell}(u) \\ \qquad\qquad\qquad\qquad\qquad\qquad - D_t^{\alpha} u_{\ell}(X,t), \ \ell \geq 0. \end{cases} \tag{6.1.12}$$

The recursive values of $u_{\ell}(X,t)$ $(\ell = 0, 1, 2, \ldots)$ can be obtained by solving (6.1.12). Eq. (6.1.8) with $p = 1$ is the original differential equation (6.1.6), and so, the approximate solution of equation (6.1.6) is given by

$$u(X,t) = u_0(X,t) + u_1(X,t) + u_2(X,t) + u_3(X,t) + \ldots. \tag{6.1.13}$$

## 6.2 Convergence analysis and error estimate [20, 206]

This sections studies the convergence of the HPM solution and the error estimate.

**Theorem 6.2.1.** *Let $\{u_n(X,t)\}_{n=0}^{\infty}$ be a sequence of a Banach space $\mathcal{B} = (\mathcal{C}(\Omega \times [0,T]), ||\cdot||)$, obtained by* (6.1.12)*, then*

*a)* $\sum_{n=0}^{\infty} u_n(X,t)$ *converges to* $u(X,t) \in \mathcal{B}$ *[solution of problem* (6.1.6)*] whenever* $\exists (0 < \sigma < 1)$ *such that* $u_n(X,t) \leq \sigma u_{(n-1)}(X,t),\ \forall\ n \in \mathbb{N}$*, and*

*b) The maximum absolute truncation error in solution* (6.1.13) *for problem* (6.1.6) *is computed as*

$$\left|\left| u(X,t) - \sum_{\imath=0}^{\ell} u_{\imath}(X,t) \right|\right| \leq \frac{\sigma^{\ell+1}}{1-\sigma} ||u_0(X,t)||. \tag{6.2.1}$$

*Proof. a)* Define the sequence $\{S_m\}_{m=0}^{\infty}$ as follows

$$\begin{cases} S_0 = u_0(X,t), \\ S_1 = u_0(X,t) + u_1(X,t), \\ \vdots \\ S_m = u_0(X,t) + u_1(X,t) + u_2(X,t) + \ldots + u_m(X,t) \end{cases}$$

To show the convergence of series (6.1.13), it is sufficient to prove that $\{S_m\}_{m=0}^{\infty}$ is Cauchy sequence.

Let $m, \ell \in \mathbb{N}$ such that $m \geq \ell$. By assumption, we get

$$\|S_m - S_{m-1}\| = \|u_m(X,t)\| \leq \sigma \|u_{m-1}(X,t)\| \leq \sigma^2 \|u_{m-2}(X,t)\| \leq \sigma^m \|u_0(X,t)\| \tag{6.2.2}$$

Using (6.2.2) and Cauchy-Schwarz inequality, we get

$$\begin{aligned} \|S_m - S_{\ell}\| = \left\| \sum_{k=0}^{m-\ell-1} (S_{\ell+k} - S_{\ell+k+1}) \right\| &\leq \sum_{k=0}^{m-\ell-1} \|S_{\ell+k} - S_{\ell+k+1}\| \leq \sum_{k=0}^{m-\ell-1} \sigma^{\ell+k+1} \|u_0(X,t)\| \\ &= \sigma^{\ell+1} \|u_0(X,t) \sum_{k=0}^{m-\ell-1} \sigma^k\| = \sigma^{\ell+1} \left( \frac{1-\sigma^{m-\ell}}{1-\sigma} \right) \|u_0(X,t)\|. \end{aligned}$$

$\because 0 < \sigma < 1$, and so, $0 < 1 - \sigma^{m-\ell} \leq 1\ \forall m \leq \ell$. Hence,

$$\|S_m - S_{\ell}\| \leq \frac{\sigma^{\ell+1}}{1-\sigma} \|u_0(X,t)\|. \tag{6.2.3}$$

Moreover $\|u_0(X,t)\|$ is bounded, and so, $\lim_{\ell\to\infty}\|S_m - S_\ell\| = 0$. This evident that $\{S_m\}_{m=0}^{\infty}$ is Cauchy sequence, and so,

$$\lim_{m\to\infty} S_m = u(X,t). \tag{6.2.4}$$

The result $b$) is direct from (6.2.3)-(6.2.4). ■

**Theorem 6.2.2.** *[20] Let $i \in \mathbb{N}$ be arbitrary. Define*

$$\lambda_i = \begin{cases} \frac{\|u_{i+1}\|}{\|u_i\|} & \text{if } \|u_i\| \neq 0, \\ 0 & \text{if } \|u_i\| = 0. \end{cases} \tag{6.2.5}$$

*In Theorem 6.2.1, $\sum_{\ell=0}^{\infty} u_\ell(X,t)$ converges to the exact solution $u(X,t)$ whenever $\exists(0 \leq \lambda_i < 1)$, and if $\lambda = \max\{\lambda_i : 0 < i \leq \ell\}$, then the maximum absolute truncation error is given by*

$\left|\left|u(X,t) - \sum_{\imath=0}^{\ell} u_\imath(X,t)\right|\right| \leq \frac{\lambda^{\ell+1}}{1-\lambda}||u_0(X,t)||$.

*Suppose $\{u_i\}_{i=0}^{\infty}, \{u_i'\}_{i=0}^{\infty}$ are obtained from two different homotopy, and $\lambda_i < \lambda_i'$ $\forall i \in \mathbb{N}$, then rate of convergence of $\sum_{\ell=0}^{\infty} u_\ell$ is larger than the rate of convergence of $\sum_{\ell=0}^{\infty} u_\ell'$.*

## 6.3 Implementation of HPM for TFCB equations

This section deals with the numerical study of the following time-fractional model of coupled viscous Burgers' equations in $(2+1)$ and $(3+1)$ dimensions by employing HPM

$$\begin{cases} D_t^\alpha U + (U\cdot\nabla)U = \nu\nabla^2 U, & \text{on } \Omega\times(0,T) \\ U(t=0) = \Psi(X), & X \in \Omega, \end{cases} \tag{6.3.1}$$

where X=(x, y), $\nabla \equiv \imath\frac{\partial}{\partial x} + \jmath\frac{\partial}{\partial y}$ in 2D (and X=(x, y, z), $\nabla \equiv \imath\frac{\partial}{\partial x} + \jmath\frac{\partial}{\partial y} + k\frac{\partial}{\partial z}$ in 3D), $U = (u,v,w) \to$ fluid vector at time $t$, $\Psi = (\psi_1,\psi_2,\psi_3)$ be the initial fluid vector, $\partial\Omega \to$boundary of $\Omega$, $\nu = \frac{1}{Re} \to$ kinematic viscosity of the flow, $Re$ is the Reynolds number.

In Cartesian co-ordinates, Equation (6.3.1) reduces to

$$\begin{cases} D_t^\alpha u + u\dfrac{\partial u}{\partial x} + v\dfrac{\partial u}{\partial y} + w\dfrac{\partial u}{\partial z} = \nu\left(\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} + \dfrac{\partial^2 u}{\partial z^2}\right), \ u(X,0) = \psi_1(X) \\ D_t^\alpha v + u\dfrac{\partial v}{\partial x} + v\dfrac{\partial v}{\partial y} + w\dfrac{\partial v}{\partial z} = \nu\left(\dfrac{\partial^2 v}{\partial x^2} + \dfrac{\partial^2 v}{\partial y^2} + \dfrac{\partial^2 v}{\partial z^2}\right), \ v(X,0) = \psi_2(X), \\ D_t^\alpha w + u\dfrac{\partial w}{\partial x} + v\dfrac{\partial w}{\partial y} + w\dfrac{\partial w}{\partial z} = \nu\left(\dfrac{\partial^2 w}{\partial x^2} + \dfrac{\partial^2 w}{\partial y^2} + \dfrac{\partial^2 w}{\partial z^2}\right), \ w(X,0) = \psi_3(X). \end{cases} \tag{6.3.2}$$

Embedding HPM on TFCB equation (6.3.2), keeping (6.1.8) in mind for $m = 1$, we get

$$\begin{cases} \dfrac{\partial u}{\partial t} = p\left[\dfrac{\partial u}{\partial t} + \nu\left(\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} + \dfrac{\partial^2 u}{\partial z^2}\right) - \left(u\dfrac{\partial u}{\partial x} + v\dfrac{\partial u}{\partial y} + w\dfrac{\partial u}{\partial z}\right) - D_t^\alpha u\right], \\ \dfrac{\partial v}{\partial t} = p\left[\dfrac{\partial v}{\partial t} + \nu\left(\dfrac{\partial^2 v}{\partial x^2} + \dfrac{\partial^2 v}{\partial y^2} + \dfrac{\partial^2 v}{\partial z^2}\right) - \left(u\dfrac{\partial v}{\partial x} + v\dfrac{\partial v}{\partial y} + w\dfrac{\partial v}{\partial z}\right) - D_t^\alpha v\right], \\ \dfrac{\partial w}{\partial t} = p\left[\dfrac{\partial w}{\partial t} + \nu\left(\dfrac{\partial^2 w}{\partial x^2} + \dfrac{\partial^2 w}{\partial y^2} + \dfrac{\partial^2 w}{\partial z^2}\right) - \left(u\dfrac{\partial w}{\partial x} + v\dfrac{\partial w}{\partial y} + w\dfrac{\partial w}{\partial z}\right) - D_t^\alpha w\right], \\ u(X,0) = \psi_1(X), \qquad v(X,0) = \psi_2(X) \qquad w(X,0) = \psi_3(X). \end{cases} \tag{6.3.3}$$

where the basic solution of Eq. (6.3.3) is assumed as

$$u(X,t) = \sum_{\ell=0}^{\infty} u_\ell(X,t)p^\ell, \quad v(X,t) = \sum_{\ell=0}^{\infty} v_\ell(X,t)p^\ell, \quad w(X,t) = \sum_{\ell=0}^{\infty} w_\ell(X,t)p^\ell, \tag{6.3.4}$$

The following is direct from (6.3.3)-(6.3.4)

$$\begin{cases} \displaystyle\sum_{\ell=0}^{\infty} p^\ell \frac{\partial u_\ell}{\partial t} = p\sum_{\ell=0}^{\infty} p^\ell \times \left[\frac{\partial u_\ell}{\partial t} + \nu\left(\frac{\partial^2 u_\ell}{\partial x^2} + \frac{\partial^2 u_\ell}{\partial y^2} + \frac{\partial^2 u_\ell}{\partial z^2}\right) - D_t^\alpha u_\ell \right. \\ \qquad \left. - \displaystyle\sum_{i=0}^{\ell}\left(u_i\frac{\partial u_{\ell-i}}{\partial x} + v_i\frac{\partial u_{\ell-i}}{\partial y} + w_i\frac{\partial u_{\ell-i}}{\partial z}\right)\right], \quad u(X,0) = \psi_1(X), \\ \displaystyle\sum_{\ell=0}^{\infty} p^\ell \frac{\partial v_\ell}{\partial t} = p\sum_{\ell=0}^{\infty} p^\ell \times \left[\frac{\partial v_\ell}{\partial t} + \nu\left(\frac{\partial^2 v_\ell}{\partial x^2} + \frac{\partial^2 v_\ell}{\partial y^2} + \frac{\partial^2 v_\ell}{\partial z^2}\right) - D_t^\alpha v_\ell \right. \\ \qquad \left. - \displaystyle\sum_{i=0}^{\ell}\left(u_i\frac{\partial v_{\ell-i}}{\partial x} + v_i\frac{\partial v_{\ell-i}}{\partial y} + w_i\frac{\partial v_{\ell-i}}{\partial z}\right)\right], \quad v(X,0) = \psi_2(X) \\ \displaystyle\sum_{\ell=0}^{\infty} p^\ell \frac{\partial w_\ell}{\partial t} = p\sum_{\ell=0}^{\infty} p^\ell \times \left[\frac{\partial w_\ell}{\partial t} + \nu\left(\frac{\partial^2 w_\ell}{\partial x^2} + \frac{\partial^2 w_\ell}{\partial y^2} + \frac{\partial^2 w_\ell}{\partial z^2}\right) - D_t^\alpha w_\ell \right. \\ \qquad \left. - \displaystyle\sum_{i=0}^{\ell}\left(u_i\frac{\partial w_{\ell-i}}{\partial x} + v_i\frac{\partial w_{\ell-i}}{\partial y} + w_i\frac{\partial w_{\ell-i}}{\partial z}\right)\right], \quad w(X,0) = \psi_3(X). \end{cases} \tag{6.3.5}$$

On equating the coefficients of the like power of $p^\ell$ in recurrence relation (6.3.5), we get the following set of linear differential equations for $u_\ell, v_\ell, w_\ell$ $(\ell = 0, 1, 2, \ldots)$ as follows:

Coefficient of $p^0$ :

$$\begin{cases} \dfrac{\partial u_0}{\partial t} = 0 \Rightarrow u_0(X,t) = \psi_1(X), \\ \dfrac{\partial v_0}{\partial t} = 0 \Rightarrow v_0(X,t) = \psi_2(X), \\ \dfrac{\partial w_0}{\partial t} = 0 \Rightarrow w_0(X,t) = \psi_3(X), \end{cases} \tag{6.3.6}$$

Coefficient of $p^{\ell+1}$ $(\ell \geq 0)$ :

$$\begin{aligned} \frac{\partial u_{\ell+1}}{\partial t} &= \frac{\partial u_\ell}{\partial t} + \nu\left(\frac{\partial^2 u_\ell}{\partial x^2} + \frac{\partial^2 u_\ell}{\partial y^2} + \frac{\partial^2 u_\ell}{\partial z^2}\right) - \sum_{i=0}^{\ell}\left(u_i\frac{\partial u_{\ell-i}}{\partial x} + v_i\frac{\partial u_{\ell-i}}{\partial y} + w_i\frac{\partial u_{\ell-i}}{\partial z}\right) - D_t^\alpha u_\ell, \\ \frac{\partial v_{\ell+1}}{\partial t} &= \frac{\partial v_\ell}{\partial t} + \nu\left(\frac{\partial^2 v_\ell}{\partial x^2} + \frac{\partial^2 v_\ell}{\partial y^2} + \frac{\partial^2 v_\ell}{\partial z^2}\right) - \sum_{i=0}^{\ell}\left(u_i\frac{\partial v_{\ell-i}}{\partial x} + v_i\frac{\partial v_{\ell-i}}{\partial y} + w_i\frac{\partial v_{\ell-i}}{\partial z}\right) - D_t^\alpha v_\ell, \\ \frac{\partial w_{\ell+1}}{\partial t} &= \frac{\partial w_\ell}{\partial t} + \nu\left(\frac{\partial^2 w_\ell}{\partial x^2} + \frac{\partial^2 w_\ell}{\partial y^2} + \frac{\partial^2 w_\ell}{\partial z^2}\right) - \sum_{i=0}^{\ell}\left(u_i\frac{\partial w_{\ell-i}}{\partial x} + v_i\frac{\partial w_{\ell-i}}{\partial y} + w_i\frac{\partial w_{\ell-i}}{\partial z}\right) - D_t^\alpha w_\ell. \end{aligned} \tag{6.3.7}$$

Now, the values of $u_\ell(X,t), v_\ell(X,t), w_\ell(X,t)$ $(\ell = 1, 2, \ldots)$ are obtained by solving recurrence relation (6.3.7) of linear differential equations by using $u_0(X,t), v_0(X,t), w_0(X,t)$ from Eq. (6.3.6).

An approximate solution of (6.3.2) is correspond to $p = 1$ in (6.3.3), that is,

$$u(X,t) = \sum_{\ell=0}^{\infty} u_\ell(X,t), \qquad v(X,t) = \sum_{\ell=0}^{\infty} v_\ell(X,t), \qquad w(X,t) = \sum_{\ell=0}^{\infty} w_\ell(X,t), \tag{6.3.8}$$

It is consequence from Theorem 6.2.2 that solutions of (6.3.8) converges to the exact solutions whenever for each sequence $\{u_m\}_{m=0}^{\infty}$, $\{v_m\}_{m=0}^{\infty}$, $\{w_m\}_{m=0}^{\infty}$, $\exists(\lambda_i)$ as defined in (6.2.1) such that $0 \leq \lambda_i < 1$.

## 6.4 Numerical results and discussion

**Example 6.4.1.** *The first case deals with the following 2D TFCB equation* (6.4.1) *with* $\psi_1(x,y) = -\sin(x+y)$, $\psi_2(x,y) = \sin(x+y)$.

$$\begin{cases} D_t^{\alpha} u + u\dfrac{\partial u}{\partial x} + v\dfrac{\partial u}{\partial y} = \nu\left(\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2}\right), \\ D_t^{\alpha} v + u\dfrac{\partial v}{\partial x} + v\dfrac{\partial v}{\partial y} = \nu\left(\dfrac{\partial^2 v}{\partial x^2} + \dfrac{\partial^2 v}{\partial y^2}\right). \end{cases} \tag{6.4.1}$$

*Embedding HPM on Eq.* (6.4.1) *with basic solutions for* $u, v$ *as defined in* (6.3.4)*, we get*

$$\begin{cases} \displaystyle\sum_{\ell=0}^{\infty} p^{\ell}\frac{\partial u_{\ell}}{\partial t} = p\sum_{\ell=0}^{\infty} p^{\ell}\left[\frac{\partial u_{\ell}}{\partial t} + \nu\left(\frac{\partial^2 u_{\ell}}{\partial x^2} + \frac{\partial^2 u_{\ell}}{\partial y^2}\right) - \sum_{i=0}^{\ell}\left(u_i\frac{\partial u_{\ell-i}}{\partial x} + v_i\frac{\partial u_{\ell-i}}{\partial y}\right) - D_t^{\alpha} u_{\ell}\right], \\ \displaystyle\sum_{\ell=0}^{\infty} p^{\ell}\frac{\partial v_{\ell}}{\partial t} = p\sum_{\ell=0}^{\infty} p^{\ell}\left[\frac{\partial v_{\ell}}{\partial t} + \nu\left(\frac{\partial^2 v_{\ell}}{\partial x^2} + \frac{\partial^2 v_{\ell}}{\partial y^2}\right) - \sum_{i=0}^{\ell}\left(u_i\frac{\partial v_{\ell-i}}{\partial x} + v_i\frac{\partial v_{\ell-i}}{\partial y}\right) - D_t^{\alpha} v_{\ell}\right], \\ u(X,0) = -\sin(x+y), \qquad v(X,0) = \sin(x+y). \end{cases} \tag{6.4.2}$$

*In* (6.4.2)*, the coefficient of* $p^0$ :

$$\begin{cases} \dfrac{\partial u_0}{\partial t} = 0 \Rightarrow u_0(X,t) = \psi_1(x,y) = -\sin(x+y), \\ \dfrac{\partial v_0}{\partial t} = 0 \Rightarrow u_0(X,t) = \psi_2(x,y) = \sin(x+y). \end{cases} \tag{6.4.3}$$

*and the coefficient of* $p^{\ell+1}$ $(\ell \geq 0)$ :

$$\begin{cases} \dfrac{\partial u_{\ell+1}}{\partial t} = \dfrac{\partial u_{\ell}}{\partial t} + \nu\left(\dfrac{\partial^2 u_{\ell}}{\partial x^2} + \dfrac{\partial^2 u_{\ell}}{\partial y^2}\right) - \displaystyle\sum_{i=0}^{\ell}\left(u_i\frac{\partial u_{\ell-i}}{\partial x} + v_i\frac{\partial u_{\ell-i}}{\partial y}\right) - D_t^{\alpha} u_{\ell}, \\ \dfrac{\partial v_{\ell+1}}{\partial t} = \dfrac{\partial v_{\ell}}{\partial t} + \nu\left(\dfrac{\partial^2 v_{\ell}}{\partial x^2} + \dfrac{\partial^2 v_{\ell}}{\partial y^2}\right) - \displaystyle\sum_{i=0}^{\ell}\left(u_i\frac{\partial v_{\ell-i}}{\partial x} + v_i\frac{\partial v_{\ell-i}}{\partial y}\right) - D_t^{\alpha} v_{\ell}. \end{cases} \tag{6.4.4}$$

*On simplifying the above equations, we get*

$$\begin{aligned} u_1 &= -v_1 = 2\nu\sin(x+y)t; \\ u_2 &= -v_2 = 2\nu\left(t - t^2\nu - \frac{t^{2-\alpha}}{\Gamma(3-\alpha)}\right)\sin(x+y); \\ u_3 &= \frac{2\nu}{3}\left(\frac{3t^{3-2\alpha}}{\Gamma(4-2\alpha)} + (3t - 6\nu t^2 + 2\nu^2 t^2) - \frac{6t^{2-\alpha}}{\Gamma(3-\alpha)} + \frac{12\nu t^{3-\alpha}}{\Gamma(4-\alpha)}\right)\sin(x+y) \\ v_3 &= -\frac{2\nu}{3}\left(\frac{3t^{3-2\alpha}}{\Gamma(4-2\alpha)} + (3t - 6\nu t^2 + 2\nu^2 t^2) - \frac{6t^{2-\alpha}}{\Gamma(3-\alpha)} + \frac{12\nu t^{3-\alpha}}{\Gamma(4-\alpha)}\right)\sin(x+y) \end{aligned}$$

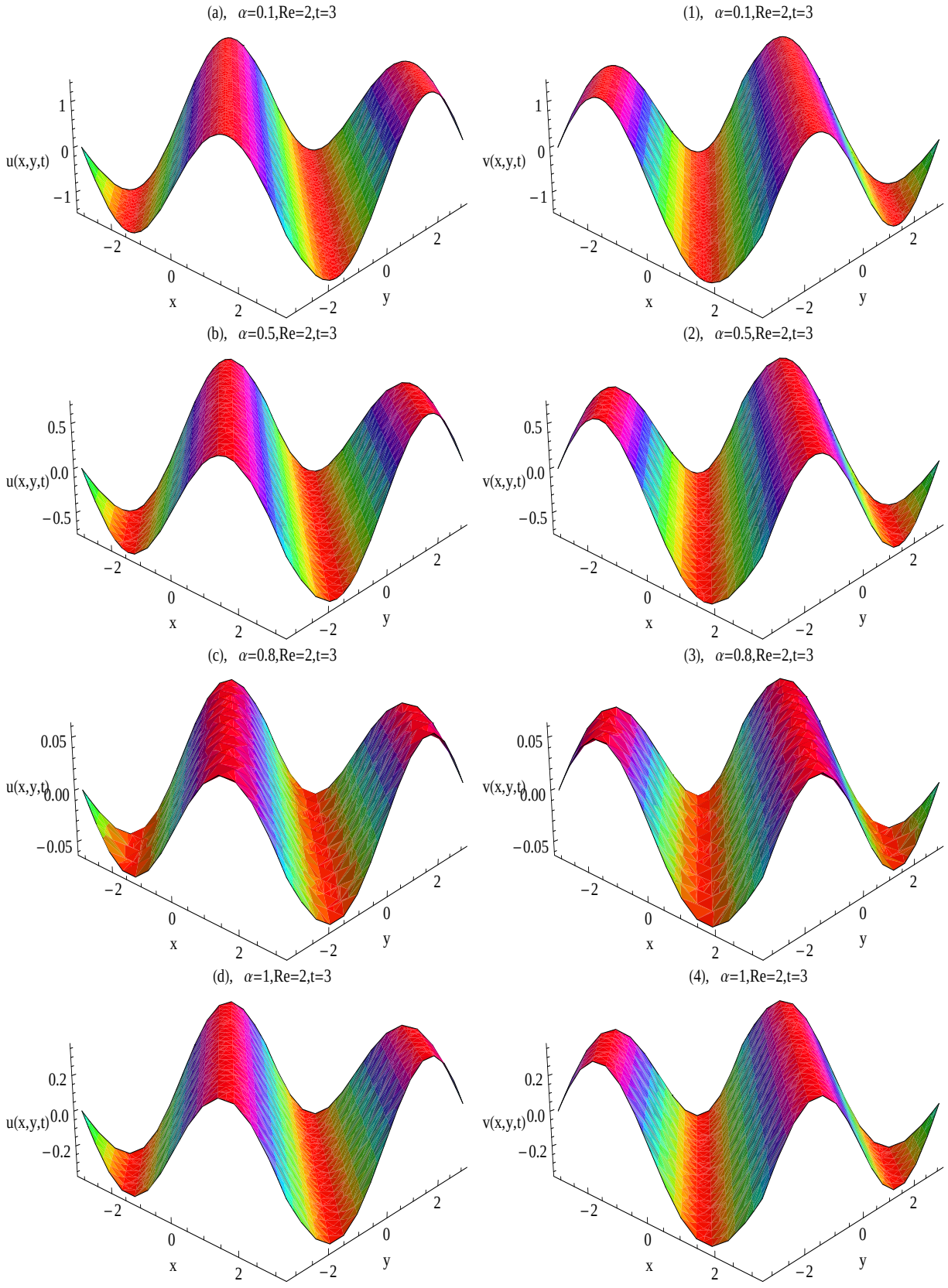


Figure 6.1: The behavior of $u, v$ of 2D TFCB equation in Example 6.4.1 with Re=2 at $t = 3$ for different values of $\alpha = 0.1, 0.5, 0.8$ and $\alpha = 1.0$ (top to bottom), respectively

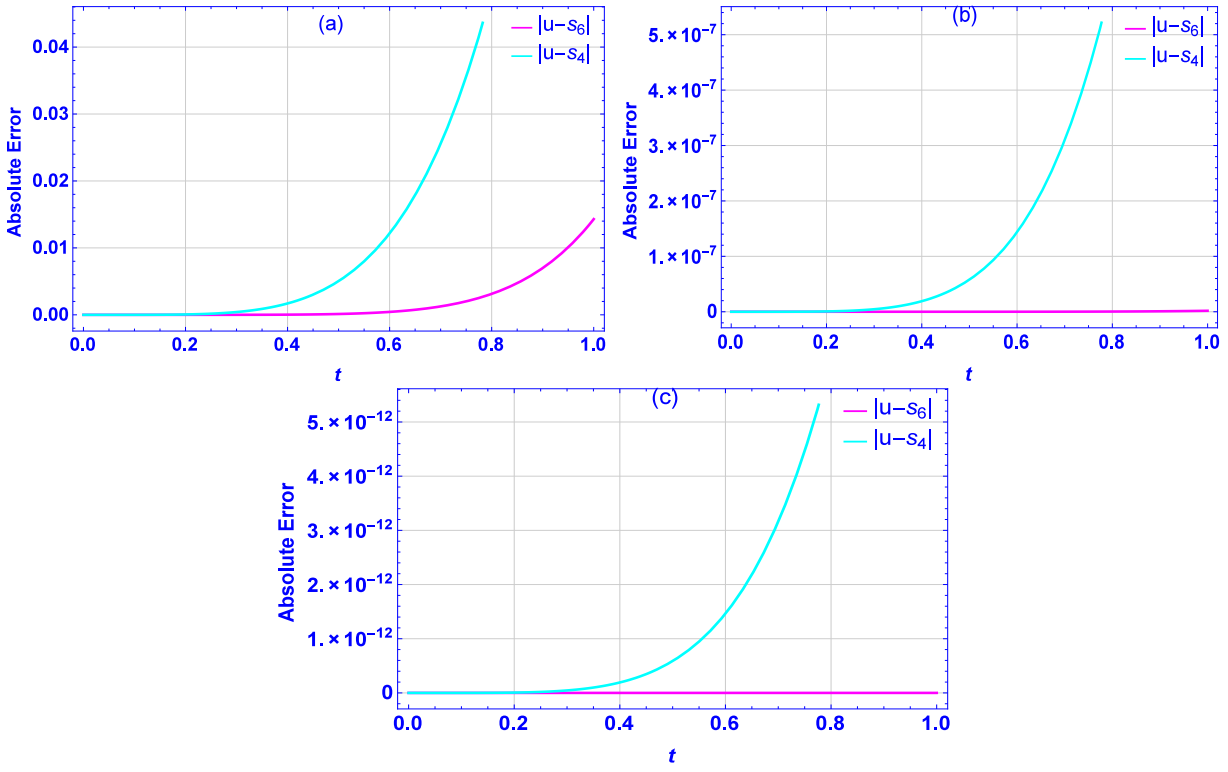


Figure 6.2: Absolute error in the solution $u$ of Ex. 6.4.1 with $x = y = \frac{\pi}{8}$ and $\alpha = 1$ at $t \leq 1$ for $(a)$ Re=1; $(b)$ Re= 10; $(c)$ Re=100

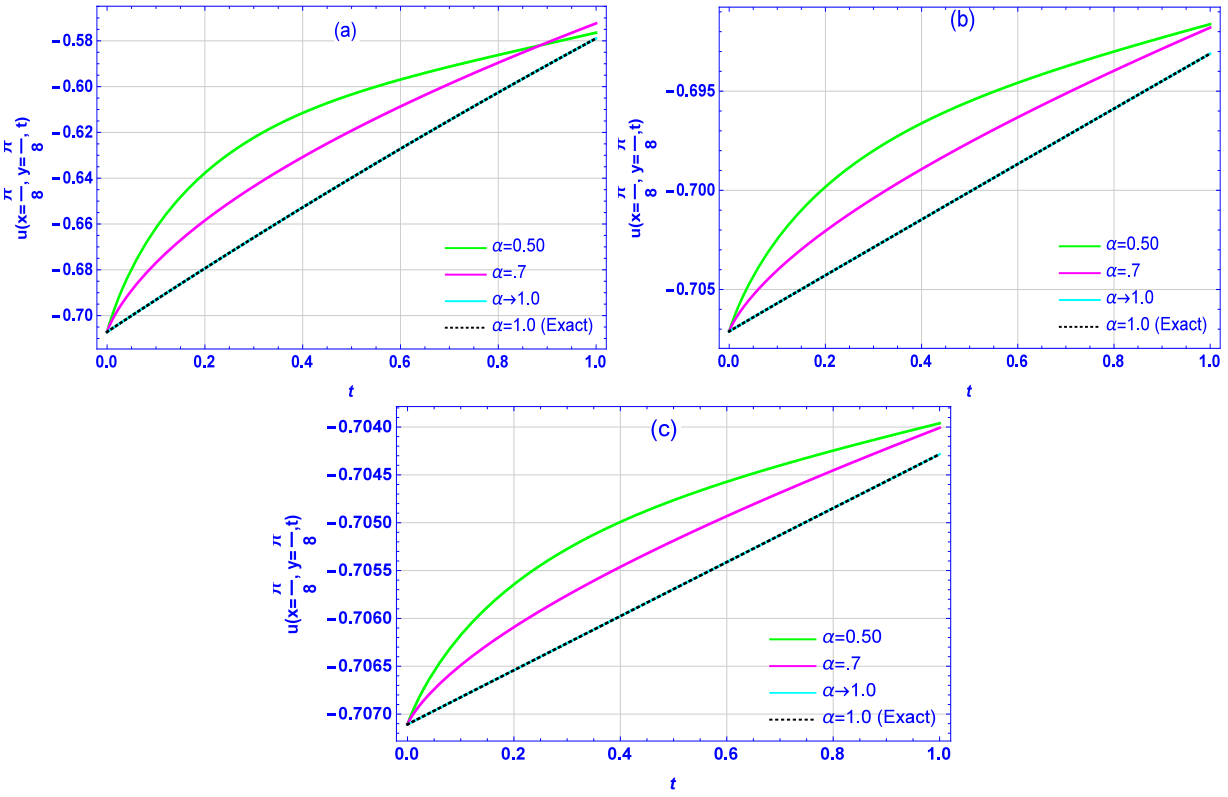


Figure 6.3: The behavior of $u$ of 2D TFCB equation in Example 6.4.1 with $x = y = \frac{\pi}{8}$ and $(a)$ Re= 1, $(b)$ Re= 10, $(c)$ Re= 500 at different time levels $t \leq 1$ considering different values of $\alpha = 0.5, 0.7, 1.0$

$$u_4 = \frac{2\nu}{3}\left\{-\frac{3t^{4-3\alpha}}{\Gamma(5-3\alpha)} + t\left(3 - 9\nu t + 6\nu^2 t^2 - \nu^3 t^3\right) + \frac{9t^{3-2\alpha}}{\Gamma(4-2\alpha)} - \frac{18\nu t^{4-2\alpha}}{\Gamma(5-2\alpha)} - \frac{9t^{2-\alpha}}{\Gamma(3-\alpha)} + \frac{36\nu t^{3-\alpha}}{\Gamma(4-\alpha)} - \frac{36\nu^2 t^{4-\alpha}}{\Gamma(5-\alpha)}\right\}\sin(x+y);$$

$$v_4 = -\frac{2\nu}{3}\left\{-\frac{3t^{4-3\alpha}}{\Gamma(5-3\alpha)} + t\left(3 - 9\nu t + 6\nu^2 t^2 - \nu^3 t^3\right) + \frac{9t^{3-2\alpha}}{\Gamma(4-2\alpha)} - \frac{18\nu t^{4-2\alpha}}{\Gamma(5-2\alpha)} - \frac{9t^{2-\alpha}}{\Gamma(3-\alpha)} + \frac{36\nu t^{3-\alpha}}{\Gamma(4-\alpha)} - \frac{36\nu^2 t^{4-\alpha}}{\Gamma(5-\alpha)}\right\}\sin(x+y); \ldots$$

*Hence, an approximate analytic solution of Burgers' equation* (6.4.1) *with first* $\ell$ *iterations is given by*

$$u(x,y,t) \approx S_\ell = \sum_{k=0}^{\ell} u_k(x,y,t), \qquad v(x,y,t) \approx \overline{S}_\ell = \sum_{k=0}^{\ell} v_k(x,y,t). \tag{6.4.5}$$

*Under the norm*

$$\|f(t)\| = \max_{0<t<1} |f(t)|,$$

*it is notice that for each sequence* $\{u_k\}_{k=0}^{\infty}$ *and* $\{v_k\}_{k=0}^{\infty}$ *the value of* $\lambda_i$ *(*$i = 1, 2, \ldots$*), as defined Theorem 6.2.2, are strictly less than unity whenever* $\nu t < \frac{1}{2}$ *and* $\nu < 1$*, e.g., for* $\{u_k\}_{k=0}^{\infty}$ *with* $\alpha = 0.5, \nu = 0.25$*:* $\lambda_0 = 0.5;\ \lambda_1 = 0.172415;\ \lambda_2 = 0.429199; \ldots$*. This confirms that the solutions* $u, v$ *in* (6.4.5) *converges to the exact solutions as* $\ell \to \infty$*.*

*The solution of Example 6.4.1 with* $\alpha = 1$ *is the closed form of*

$$\begin{aligned} u(x,y,t) &= -\sin(x+y)e^{-2\nu t}, \\ v(x,y,t) &= \sin(x+y)e^{-2\nu t}, \end{aligned} \tag{6.4.6}$$

*which is the solution of classical coupled viscous Burgers' equation. The solution behavior of TFCB equation* (6.4.5) *with* $\alpha = 0.1, 0.5, 0.8, 1$ *and Re*$= 2$ *is depicted in Figure 6.1. Similar behavior has been found in [248]. It is worth mentioning from Table 6.1 that HPM solutions* $u, v$ *are agreed excellently with the solutions obtained by FRDTM [248], and the errors in* $u, v$ *decreases rapidly as iterations/terms increases. Table 6.2 and Figure 6.2 show that for a given* $t$*, the approximate solution* $u$ *is more accurate for large Reynolds number*

Table 6.1: Comparison of HPM solutions with FRDTM solutions and exact solutions of Example 6.4.1 with $\alpha = 1, Re = \frac{1}{\nu} = 2$ considering first $n$ terms ($n = 5, 7$)

| | | | HPM | | | | FRDTM [248] | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $x$ | $y$ | $t$ | $\lvert u - S_4\rvert$ | $\lvert u - S_6\rvert$ | $\lvert v - \overline{S}_4\rvert$ | $\lvert v - \overline{S}_6\rvert$ | $\lvert u - S_4\rvert$ | $\lvert u - S_6\rvert$ | $\lvert v - \overline{S}_4\rvert$ | $\lvert v - \overline{S}_6\rvert$ |
| $\frac{\pi}{8}$ | $\frac{\pi}{16}$ | 0.2 | 1.43351E-06 | 1.37647E-09 | 1.43351E-06 | 1.37647E-09 | 1.43351E-06 | 1.37647E-09 | 1.43000E-06 | 1.37647E-09 |
| | | 0.4 | 4.44200E-05 | 1.71960E-07 | 4.44200E-05 | 1.71960E-07 | 4.44200E-05 | 1.71960E-07 | 4.44200E-05 | 1.71960E-07 |
| | | 0.6 | 3.26878E-04 | 2.86891E-06 | 3.26878E-04 | 2.86891E-06 | 3.26878E-04 | 2.86891E-06 | 3.26878E-04 | 2.86891E-06 |
| | | 0.8 | 1.33580E-03 | 2.09958E-05 | 1.33580E-03 | 2.09958E-05 | 1.33580E-03 | 2.09958E-05 | 1.33580E-03 | 2.09958E-05 |
| | $\frac{\pi}{8}$ | 0.2 | 1.82452E-06 | 1.75191E-09 | 1.82452E-06 | 1.75191E-09 | 1.82452E-06 | 1.75191E-09 | 1.82452E-06 | 1.75191E-09 |
| | | 0.4 | 5.65360E-05 | 2.18864E-07 | 5.65360E-05 | 2.18864E-07 | 5.65360E-05 | 2.18864E-07 | 5.65360E-05 | 2.18864E-07 |
| | | 0.6 | 4.16036E-04 | 3.65143E-06 | 4.16036E-04 | 3.65143E-06 | 4.16036E-04 | 3.65143E-06 | 4.16036E-04 | 3.65143E-06 |
| | | 0.8 | 1.70015E-03 | 2.67226E-05 | 1.70015E-03 | 2.67226E-05 | 1.70015E-03 | 2.67226E-05 | 1.70015E-03 | 2.67226E-05 |
| | $\frac{\pi}{4}$ | 0.2 | 2.38385E-06 | 2.28898E-09 | 2.38385E-06 | 2.28898E-09 | 2.38385E-06 | 2.28898E-09 | 2.38385E-06 | 2.28898E-09 |
| | | 0.4 | 7.38678E-05 | 2.85959E-07 | 7.38678E-05 | 2.85959E-07 | 7.38678E-05 | 2.85959E-07 | 7.38678E-05 | 2.85959E-07 |
| | | 0.6 | 5.43577E-04 | 4.77083E-06 | 5.43577E-04 | 4.77083E-06 | 5.43577E-04 | 4.77083E-06 | 5.43577E-04 | 4.77083E-06 |
| | | 0.8 | 2.22135E-03 | 3.49147E-05 | 2.22135E-03 | 3.49147E-05 | 2.22135E-03 | 3.49147E-05 | 2.22135E-03 | 3.49147E-05 |
| $\frac{\pi}{4}$ | $\frac{\pi}{16}$ | 0.2 | 2.14540E-06 | 2.06003E-09 | 2.14540E-06 | 2.06003E-09 | 2.14540E-06 | 2.06003E-09 | 2.14540E-06 | 2.06003E-09 |
| | | 0.4 | 6.64793E-05 | 2.57356E-07 | 6.64793E-05 | 2.57356E-07 | 6.64793E-05 | 2.57356E-07 | 6.64793E-05 | 2.57356E-07 |
| | | 0.6 | 4.89207E-04 | 4.29363E-06 | 4.89207E-04 | 4.29363E-06 | 4.89207E-04 | 4.29363E-06 | 4.89207E-04 | 4.29363E-06 |
| | | 0.8 | 1.99916E-03 | 3.14224E-05 | 1.99916E-03 | 3.14224E-05 | 1.99916E-03 | 3.14224E-05 | 1.99916E-03 | 3.14224E-05 |
| | $\frac{\pi}{8}$ | 0.2 | 2.38385E-06 | 2.28898E-09 | 2.38385E-06 | 2.28898E-09 | 2.38385E-06 | 2.28898E-09 | 2.38385E-06 | 2.28898E-09 |
| | | 0.4 | 7.38678E-05 | 2.85959E-07 | 7.38678E-05 | 2.85959E-07 | 7.38678E-05 | 2.85959E-07 | 7.38678E-05 | 2.85959E-07 |
| | | 0.6 | 5.43577E-04 | 4.77083E-06 | 5.43577E-04 | 4.77083E-06 | 5.43577E-04 | 4.77083E-06 | 5.43577E-04 | 4.77083E-06 |
| | | 0.8 | 2.22135E-03 | 3.49147E-05 | 2.22135E-03 | 3.49147E-05 | 2.22135E-03 | 3.49147E-05 | 2.22135E-03 | 3.49147E-05 |
| | $\frac{\pi}{4}$ | 0.2 | 2.58026E-06 | 2.47758E-09 | 2.58026E-06 | 2.47758E-09 | 2.58026E-06 | 2.47757E-09 | 2.58026E-06 | 2.47757E-09 |
| | | 0.4 | 7.99540E-05 | 3.09520E-07 | 7.99540E-05 | 3.09520E-07 | 7.99540E-05 | 3.09520E-07 | 7.99540E-05 | 3.09520E-07 |
| | | 0.6 | 5.88364E-04 | 5.16391E-06 | 5.88364E-04 | 5.16391E-06 | 5.88364E-04 | 5.16391E-06 | 5.88364E-04 | 5.16391E-06 |
| | | 0.8 | 2.40437E-03 | 3.77914E-05 | 2.40437E-03 | 3.77914E-05 | 2.40437E-03 | 3.77914E-05 | 2.40437E-03 | 3.77914E-05 |

*(Re $\geq$ 100). The solution behavior of $u$ for different $\alpha = 0.5, 0.7, 1.0$ has been depicted in Figure 6.3 for Re $= 1, 10, 500$. Similar behavior can be found for $v$. The findings confirm that HPM solutions $u, v$ converges to exact solutions comparatively more fast for large Reynolds numbers.*

**Example 6.4.2.** *The second case deals with 2D TFCB equation* (6.4.1) *with* $\psi_1(x,y) = -e^{x+y}$, $\psi_2(x,y) = e^{x+y}$.

*Similar to Example 6.4.1, the values of* $u_\ell(x,y,t), v_\ell(x,y,t)$ *can be obtained by solving the recurrence relation* (6.4.3)-(6.4.4) *with* $\psi_1(x,y,t) = -e^{x+y}, \psi_2(x,y,t) = e^{x+y}$ *as follows:*

$$\begin{aligned}
u_0(x,y,t) &= -v_0(x,y,t) = -e^{x+y};\\
u_1(x,y,t) &= -v_1(x,y,t) = -2\nu t e^{x+y};\\
u_2(x,y,t) &= -v_2(x,y,t) = -2\nu\left(t + \nu t^2 - \frac{t^{2-\alpha}}{\Gamma(3-\alpha)}\right)e^{x+y};\\
u_3(x,y,t) &= -v_3(x,y,t)\\
&= -\frac{2\nu}{3}\left(\frac{3t^{3-2\alpha}}{\Gamma(4-2\alpha)} + t(3 + 6\nu t + 2\nu^2 t^2) + \frac{6t^{2-\alpha}}{\Gamma(3-\alpha)} + \frac{12\nu t^{3-\alpha}}{\Gamma(4-\alpha)}\right)e^{x+y};\\
u_4(x,y,t) &= -v_4(x,y,t) = \frac{2\nu}{3}e^{x+y}\left(\frac{3t^{4-3\alpha}}{\Gamma(5-3\alpha)} - t(3 + 9\nu t + 6\nu^2 t^2 + \nu^3 t^3)\right.\\
&\quad \left. - \frac{9t^{3-2\alpha}}{\Gamma(4-2\alpha)} - \frac{18\nu t^{4-2\alpha}}{\Gamma(5-\alpha)} + \frac{9t^{2-\alpha}}{\Gamma(3-\alpha)} + \frac{36\nu t^{3-\alpha}}{\Gamma(4-\alpha)} + \frac{36\nu^2 t^{4-\alpha}}{\Gamma(5-\alpha)}\right), \ldots
\end{aligned} \tag{6.4.7}$$

*The approximate analytic solution of Burgers' equation with first $\ell$ iterations is given by*

$$u(x,y,t) \approx S_\ell = \sum_{k=0}^{\ell} u_k(x,y), \qquad v(x,y,t) \approx \overline{S}_\ell = \sum_{k=0}^{\ell} v_k(x,y,t), \tag{6.4.8}$$

*and which converges to the exact solution as $\ell \to \infty$, and the solution of Example 6.4.2 with $\alpha = 1$ is the closed form of*

$$u(x,y,t) = -e^{x+y+2\nu t}; \qquad v(x,y,t) = e^{x+y+2\nu t}, \tag{6.4.9}$$

*which is the solution of classical coupled viscous Burgers' equation. The solution behavior of TFCB equation is depicted in Figure 6.4 for $\alpha = 0.1,\ 0.8, 1.0$ and Re$=$ 2. Similar behavior has been found in [248]. It is evident from Table 6.4 that HPM solutions $u, v$*

Table 6.2: Absolute errors in first five terms approximation of HPM solutions of $(u, v)$ in Example 6.4.1 with $\alpha = 1, \mathrm{Re} = 10, 100, 500$ at different time levels $t \leq 1$

| For $u$ | | $t = 0.25$ | | | $t = 0.5$ | | | $t = 1.0$ | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $x$ | $y$ | Re=10 | Re=100 | Re=500 | Re=10 | Re=100 | Re=500 | Re=10 | Re=100 | Re=500 |
| $\frac{\pi}{8}$ | $\frac{\pi}{16}$ | 1.43483E-09 | 1.44329E-14 | 0 | 4.55368E-08 | 4.62186E-13 | 1.11022E-16 | 1.43351E-06 | 1.47660E-11 | 4.66294E-15 |
| | $\frac{\pi}{8}$ | 1.82619E-09 | 1.84297E-14 | 0 | 5.79573E-08 | 5.88196E-13 | 2.22045E-16 | 1.82452E-06 | 1.87935E-11 | 5.99520E-15 |
| | $\frac{\pi}{4}$ | 2.38603E-09 | 2.40918E-14 | 0 | 7.57249E-08 | 7.68496E-13 | 2.22045E-16 | 2.38385E-06 | 2.45549E-11 | 7.88258E-15 |
| $\frac{\pi}{4}$ | $\frac{\pi}{16}$ | 2.14737E-09 | 2.16494E-14 | 0 | 6.81506E-08 | 6.91669E-13 | 2.22045E-16 | 2.14540E-06 | 2.20989E-11 | 7.10543E-15 |
| | $\frac{\pi}{8}$ | 2.38603E-09 | 2.40918E-14 | 0 | 7.57249E-08 | 7.68496E-13 | 2.22045E-16 | 2.38385E-06 | 2.45549E-11 | 7.88258E-15 |
| | $\frac{\pi}{4}$ | 2.58262E-09 | 2.59792E-14 | 1.11020E-16 | 8.19640E-08 | 8.31890E-13 | 2.22045E-16 | 2.58026E-06 | 2.65781E-11 | 8.54872E-15 |
| For $v$ | | | | | | | | | | |
| $\frac{\pi}{8}$ | $\frac{\pi}{16}$ | 1.43483E-09 | 1.44329E-14 | 0 | 4.55368E-08 | 4.62186E-13 | 1.11022E-16 | 1.43351E-06 | 1.47661E-11 | 4.66294E-15 |
| | $\frac{\pi}{8}$ | 1.82619E-09 | 1.84297E-14 | 0 | 5.79573E-08 | 5.88196E-13 | 2.22045E-16 | 1.82452E-06 | 1.87936E-11 | 6.10623E-15 |
| | $\frac{\pi}{4}$ | 2.38603E-09 | 2.40918E-14 | 0 | 7.57249E-08 | 7.68496E-13 | 2.22045E-16 | 2.38385E-06 | 2.45549E-11 | 7.88258E-15 |
| $\frac{\pi}{4}$ | $\frac{\pi}{16}$ | 2.14737E-09 | 2.16494E-14 | 0 | 6.81506E-08 | 6.91669E-13 | 2.22045E-16 | 2.14540E-06 | 2.20989E-11 | 7.10543E-15 |
| | $\frac{\pi}{8}$ | 2.38603E-09 | 2.40918E-14 | 0 | 7.57249E-08 | 7.68496E-13 | 2.22045E-16 | 2.38385E-06 | 2.45549E-11 | 7.88258E-15 |
| | $\frac{\pi}{4}$ | 2.58262E-09 | 2.59792E-14 | 1.11020E-16 | 8.19640E-08 | 8.31890E-13 | 2.22045E-16 | 2.58026E-06 | 2.65781E-11 | 8.54872E-15 |

Table 6.3: Absolute errors in first five terms approximation of HPM solution $u$ in Example 6.4.2 with $\alpha = 1, \mathrm{Re} = 10, 100, 500$ at different time levels $t \leq 1$

| For $u$ | | $t = 0.25$ | | | $t = 0.5$ | | | $t = 1.0$ | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $y$ | $x$ | Re=10 | Re=100 | Re=500 | Re=10 | Re=100 | Re=500 | Re=10 | Re=100 | Re=500 |
| 0.25 | 0.25 | 4.32958E-09 | 4.26326E-14 | 0 | 1.39716E-07 | 1.37623E-12 | 2.22045E-16 | 4.54744E-06 | 4.41129E-11 | 1.39888E-14 |
| | 0.50 | 5.55929E-09 | 5.55112E-14 | 0 | 1.79399E-07 | 1.76659E-12 | 8.88178E-16 | 5.83903E-06 | 5.66418E-11 | 1.82077E-14 |
| | 0.75 | 7.13827E-09 | 7.10543E-14 | 0 | 2.30353E-07 | 2.26885E-12 | 8.88178E-16 | 7.49746E-06 | 7.27298E-11 | 2.35367E-14 |
| 0.50 | 0.25 | 5.55929E-09 | 5.55112E-14 | 0 | 1.79399E-07 | 1.76659E-12 | 8.88178E-16 | 5.83903E-06 | 5.66418E-11 | 1.82077E-14 |
| | 0.50 | 7.13827E-09 | 7.10543E-14 | 0 | 2.30353E-07 | 2.26885E-12 | 8.88178E-16 | 7.49746E-06 | 7.27298E-11 | 2.35367E-14 |
| | 0.75 | 9.16572E-09 | 9.10383E-14 | 4.44E-16 | 2.95780E-07 | 2.91323E-12 | 8.88178E-16 | 9.62692E-06 | 9.33866E-11 | 2.97540E-14 |

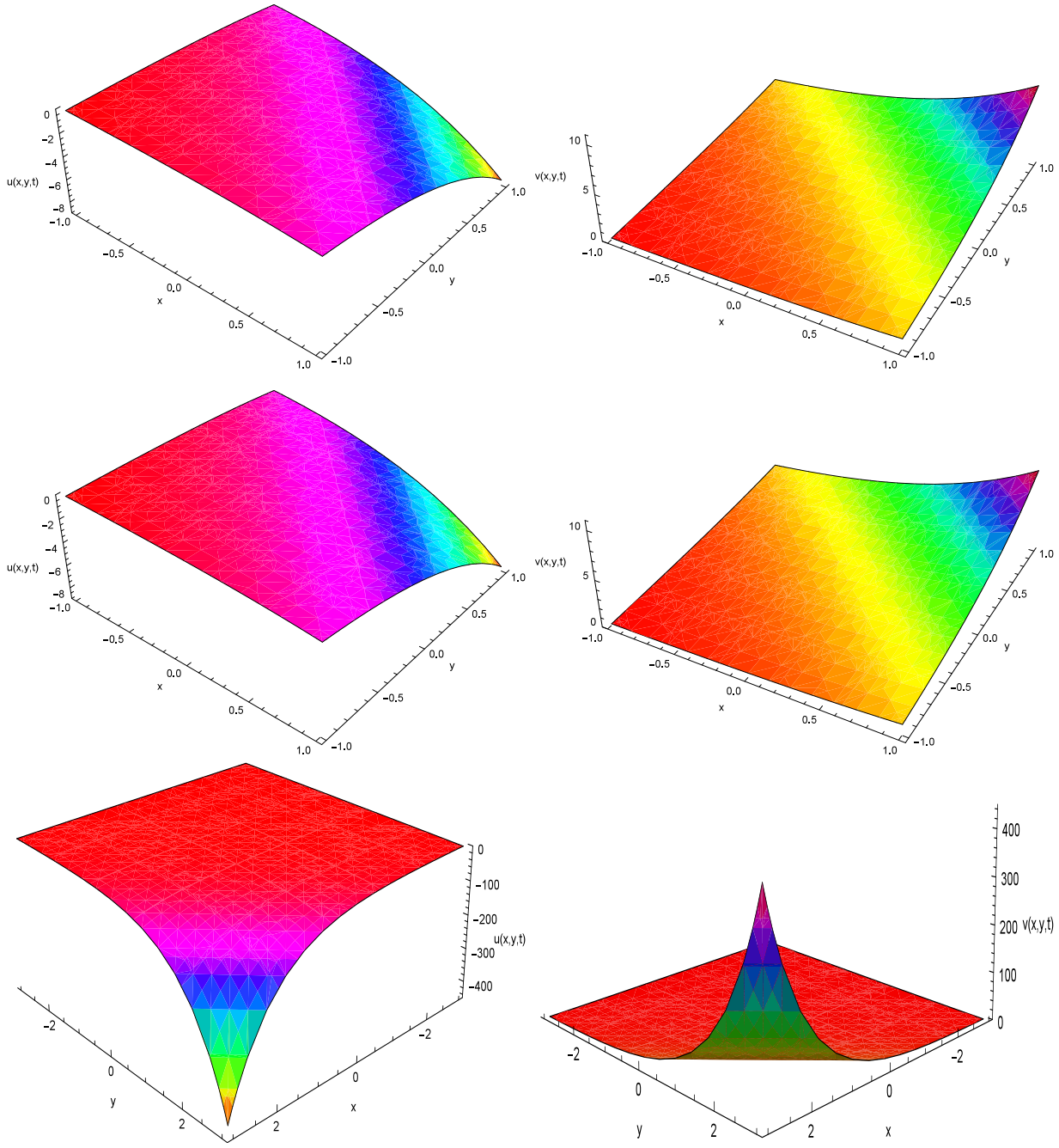


Figure 6.4: Behavior of $u$ and $v$ of 2D TFCB equation in Example 6.4.2 with Re= 2 at $t = 0.05$ for $\alpha = 0.1, 0.8, 1.0$ (top to bottom)

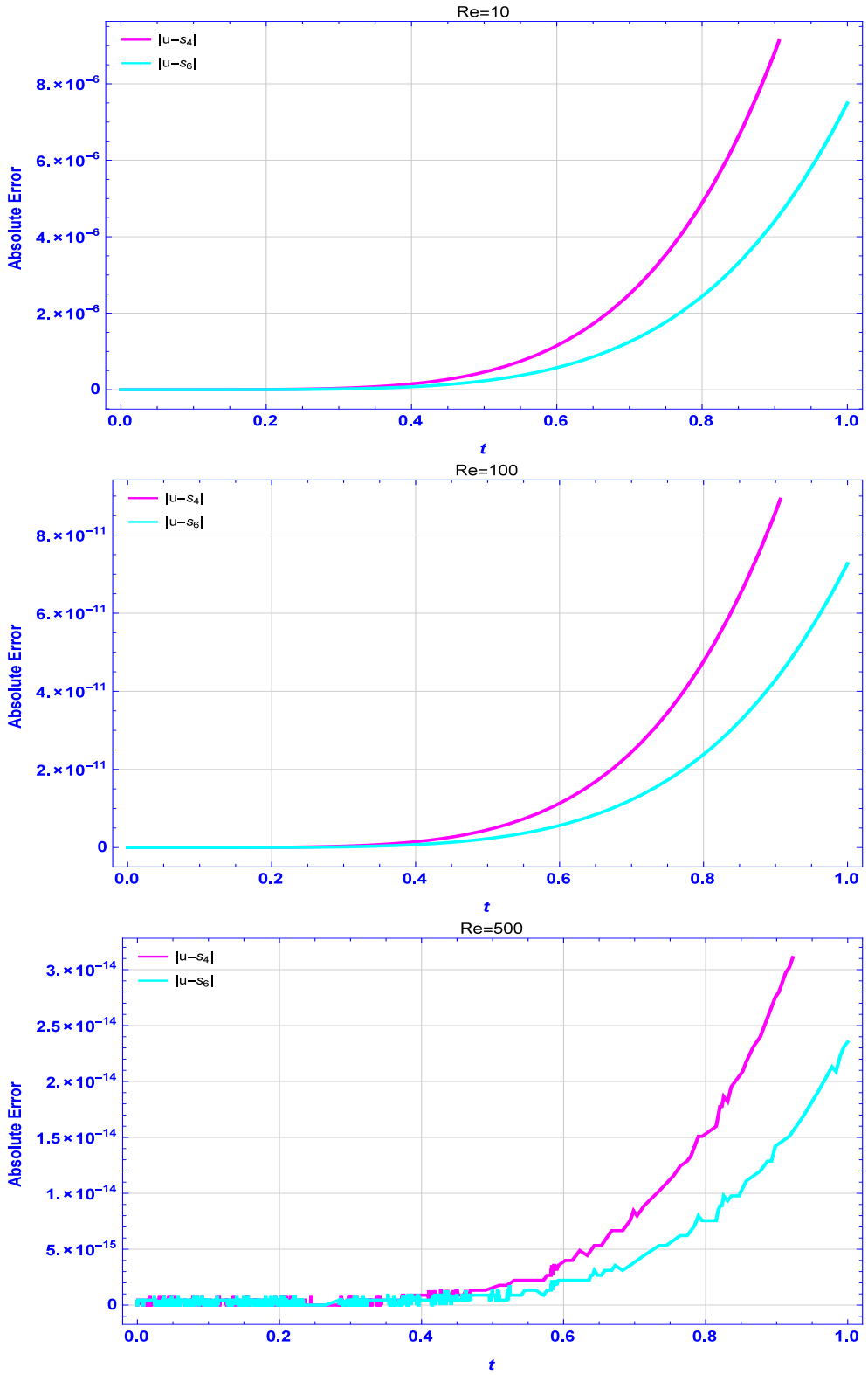


Figure 6.5: Absolute error in the solution $u$ of Example 6.4.2 with $\alpha = 1$ at $t \leq 1$ for $x = y = 0.5$ and different values of Re$= 10, 100, 500$

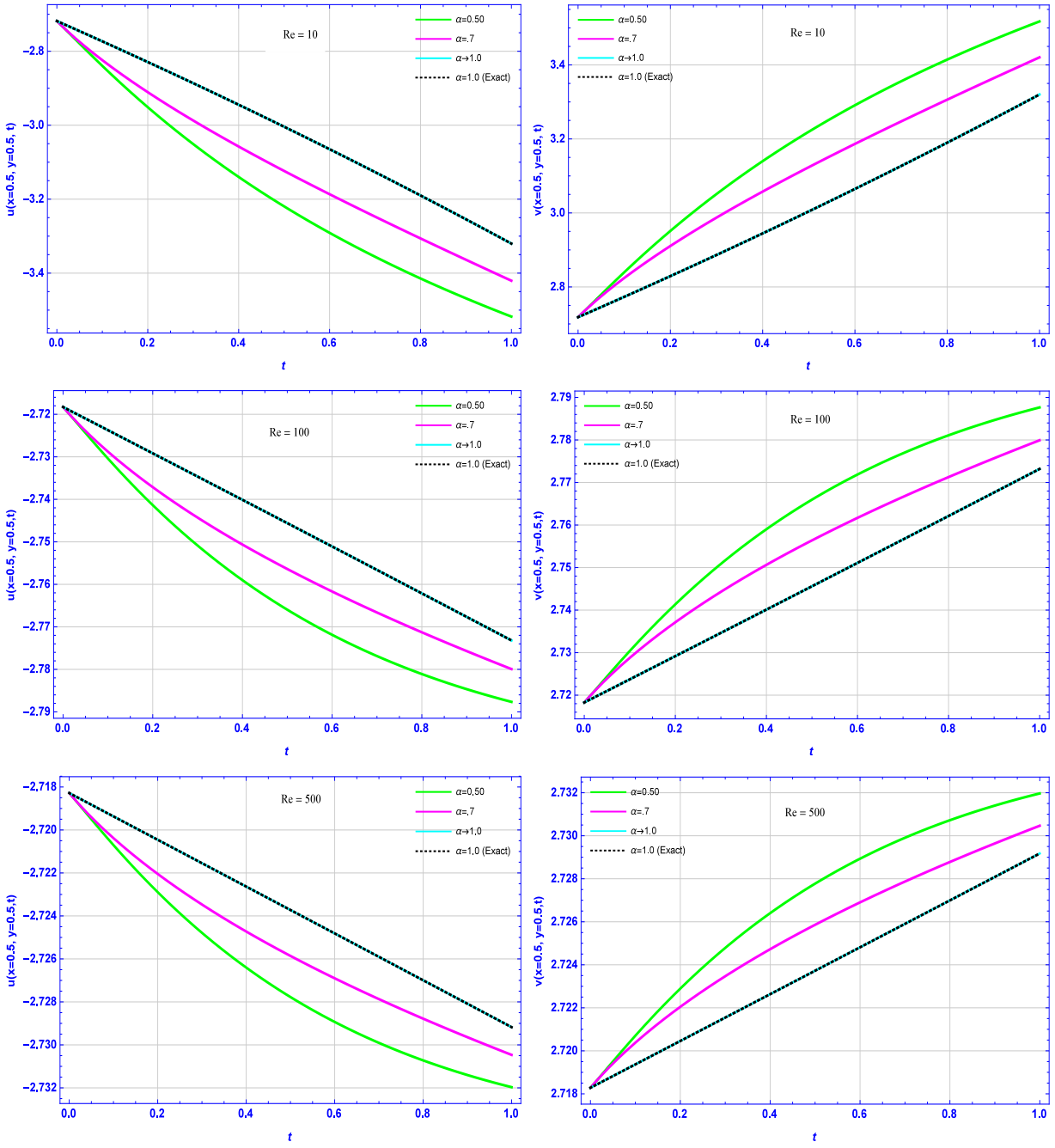


Figure 6.6: The behavior of $u$ and $v$ of 2D TFCB equation in Example 6.4.2 with $x = y = 0.5$ and different values of Re$= 10, 100, 500$ at different time levels $t \leq 1$ for different values of $\alpha = 0.5, 0.7, 1.0$

Table 6.4: Comparison of HPM solutions with FRDTM [248] solutions and exact solutions of Example 6.4.2 with $\alpha = 1, Re = \frac{1}{\nu} = 2$ considering first $n$ terms ($n = 5, 7$)

| $y = 0.10$ | | Absolute Error in $u$ | | | Absolute Error in $v$ | | |
|---|---|---|---|---|---|---|---|
| | | FRDTM [248] | HPM | | FRDTM [248] | HPM | |
| $x$ | $t$ | $\lvert u - S_4\rvert$ | $\lvert u - S_4\rvert$ | $\lvert u - S_6\rvert$ | $\lvert v - \overline{S}_4\rvert$ | $\lvert v - \overline{S}_4\rvert$ | $\lvert v - \overline{S}_6\rvert$ |
| 0.2 | 0.1 | 1.14390E-07 | 1.14390E-07 | 3.77673E-09 | 1.14390E-07 | 1.14390E-07 | 3.77673E-09 |
| | 0.2 | 3.72313E-06 | 3.72313E-06 | 2.43491E-07 | 3.72313E-06 | 3.72313E-06 | 2.43491E-07 |
| | 0.3 | 2.87622E-05 | 2.87622E-05 | 2.79431E-06 | 2.87622E-05 | 2.87622E-05 | 2.79431E-06 |
| 0.4 | 0.1 | 1.39716E-07 | 1.39716E-07 | 4.61291E-09 | 1.39716E-07 | 1.39716E-07 | 4.61291E-09 |
| | 0.2 | 4.54744E-06 | 4.54744E-06 | 2.97400E-07 | 4.54744E-06 | 4.54744E-06 | 2.97400E-07 |
| | 0.3 | 3.51303E-05 | 3.51303E-05 | 3.41298E-06 | 3.51303E-05 | 3.51303E-05 | 3.41298E-06 |
| 0.6 | 0.1 | 1.70650E-07 | 1.70650E-07 | 5.63422E-09 | 1.70650E-07 | 1.70650E-07 | 5.63422E-09 |
| | 0.2 | 5.55425E-06 | 5.55425E-06 | 3.63246E-07 | 5.55425E-06 | 5.55425E-06 | 3.63246E-07 |
| | 0.3 | 4.29082E-05 | 4.29082E-05 | 4.16862E-06 | 4.29082E-05 | 4.29082E-05 | 4.16862E-06 |
| $y = 0.50$ | | | | | | | |
| 0.2 | 0.1 | 1.70650E-07 | 1.70650E-07 | 5.63422E-09 | 1.70650E-07 | 1.70650E-07 | 5.63422E-09 |
| | 0.2 | 5.55425E-06 | 5.55425E-06 | 3.63246E-07 | 5.55425E-06 | 5.55425E-06 | 3.63246E-07 |
| | 0.3 | 4.29082E-05 | 4.29082E-05 | 4.16862E-06 | 4.29082E-05 | 4.29082E-05 | 4.16862E-06 |
| 0.4 | 0.1 | 2.08432E-07 | 2.08432E-07 | 6.88165E-09 | 2.08432E-07 | 2.08432E-07 | 6.88165E-09 |
| | 0.2 | 6.78398E-06 | 6.78398E-06 | 4.43669E-07 | 6.78398E-06 | 6.78398E-06 | 4.43669E-07 |
| | 0.3 | 5.24082E-05 | 5.24082E-05 | 5.09157E-06 | 5.24082E-05 | 5.24082E-05 | 5.09157E-06 |
| 0.6 | 0.1 | 2.54580E-07 | 2.54580E-07 | 8.40526E-09 | 2.54580E-07 | 2.54580E-07 | 8.40526E-09 |
| | 0.2 | 8.28597E-06 | 8.28597E-06 | 5.41899E-07 | 8.28597E-06 | 8.28597E-06 | 5.41899E-07 |
| | 0.3 | 6.40115E-05 | 6.40115E-05 | 6.21885E-06 | 6.40115E-05 | 6.40115E-05 | 6.21885E-06 |

*are agreed excellently with the solutions obtained by FRDTM [248], and the errors in $u, v$ decreases rapidly as iterations/terms increases. The absolute error in u for different values of Reynolds number (Re=$10, 100, 500$) and $\alpha = 0.5, 0.7, 1$ is reported in Table 6.3, and similar behavior of absolute error can be found for $v$. From Table 6.3 and Figure 6.5, it is evident that for a given $t$, the approximate solutions are more accurate for large Reynolds number ($Re \geq 100$). The solution behavior of $u, v$ for different $\alpha = 0.5, 0.7, 1.0$ has been depicted in Figure 6.6 for $Re = 1, 10, 500$. The findings confirm that HPM solutions $u, v$ converges to exact solutions comparatively more fast for large Reynolds numbers.*

**Example 6.4.3.** *The third case deals with initial value 3D TFCB equation* (6.3.2) *with* $\psi_1(X) = -0.5x + y + z, \quad \psi_2(X) = x - 0.5y + z, \quad \psi_3(X) = x + y - 0.5z, \ X = (x, y, z)$. *Similar to the previous problems, the recursive values of* $u_\ell(X, t) \equiv u_\ell, v_\ell(X, t) \equiv$

*$v_\ell, w_\ell(X,t) \equiv w_\ell$ $(\ell \geq 0)$ are obtained by solving* (6.3.6)-(6.3.7) *as follows:*

$$u_1 = -2.25xt; \qquad v_1 = -2.25yt; \qquad w_1(x,y,z,t) = -2.25zt;$$

$$u_2 = -2.25xt - 1.125t^2x + 2.25t^2y + 2.25t^2z + \frac{t^{2-\alpha}(2.25x)}{\Gamma(3-\alpha)};$$

$$v_2 = 2.25t^2x - 2.25ty - 1.125t^2y + 2.25t^2z + \frac{t^{2-\alpha}(2.25y)}{\Gamma(3-\alpha)};$$

$$w_2 = 2.25t^2x + 2.25t^2y - 2.25tz - 1.125t^2z + \frac{t^{2-\alpha}(2.25z)}{\Gamma(3-\alpha)};$$

$$u_3 = 4.5t^2(y+z) - 2.25xt(1+t) - 5.0625t^3x + \frac{4.5t^{2-\alpha}}{\Gamma(3-\alpha)} - 4.5t^{3-\alpha}\frac{-x+2y+2z}{\Gamma(4-\alpha)} - \frac{2.25xt^{3-2\alpha}}{\Gamma(4-2\alpha)};$$

$$v_3 = 4.5t^2(x+z) - 2.25yt(1+t) - 5.0625t^3y + \frac{4.5t^{2-\alpha}}{\Gamma(3-\alpha)} - 4.5t^{3-\alpha}\frac{2x-y+2z}{\Gamma(4-\alpha)} - \frac{2.25yt^{3-2\alpha}}{\Gamma(4-2\alpha)};$$

$$w_3 = 4.5t^2(x+y) - 2.25zt(1+t) - 5.0625t^3z + \frac{4.5t^{2-\alpha}}{\Gamma(3-\alpha)} - 4.5t^{3-\alpha}\frac{2x+2y-z}{\Gamma(4-\alpha)} - \frac{2.25zt^{3-2\alpha}}{\Gamma(4-2\alpha)}; \ldots$$

*The approximate analytic solution with first four terms $\ell = 3$ is given by*

$$\begin{aligned} u(x,y,z,t) &= \sum_{k=0}^{\ell} u_k(x,y,z,t), \\ v(x,y,z,t) &= \sum_{k=0}^{\ell} v_k(x,y,z,t), \\ w(x,y,z,t) &= \sum_{k=0}^{\ell} w_k(x,y,z,t), \end{aligned} \tag{6.4.10}$$

*which converges to the exact solution as $\ell \to \infty$. Moreover, the solution of Example 6.4.3 with $\alpha = 1$ is the closed form of*

$$\begin{aligned} u(x,y,z,t) &= \frac{-0.5x + y + z - 2.25xt}{1 - 2.25t^2}; \\ v(x,y,z,t) &= \frac{x - 0.5y + z - 2.25yt}{1 - 2.25t^2}, \\ w(x,y,z,t) &= \frac{x + y - 0.5z - 2.25zt}{1 - 2.25t^2}, \end{aligned}$$

*which is the solution of classical initial value coupled viscous Burgers' equation. The behavior of velocity profile $(u,v,w)$ at $t = 0.1$ is depicted in Fig. 6.7 for $z = 0.5$ and $\alpha = 0.5, 1.0$. The similar velocity profile has been obtained in [248] by FRDTM. Also, the solution be-*

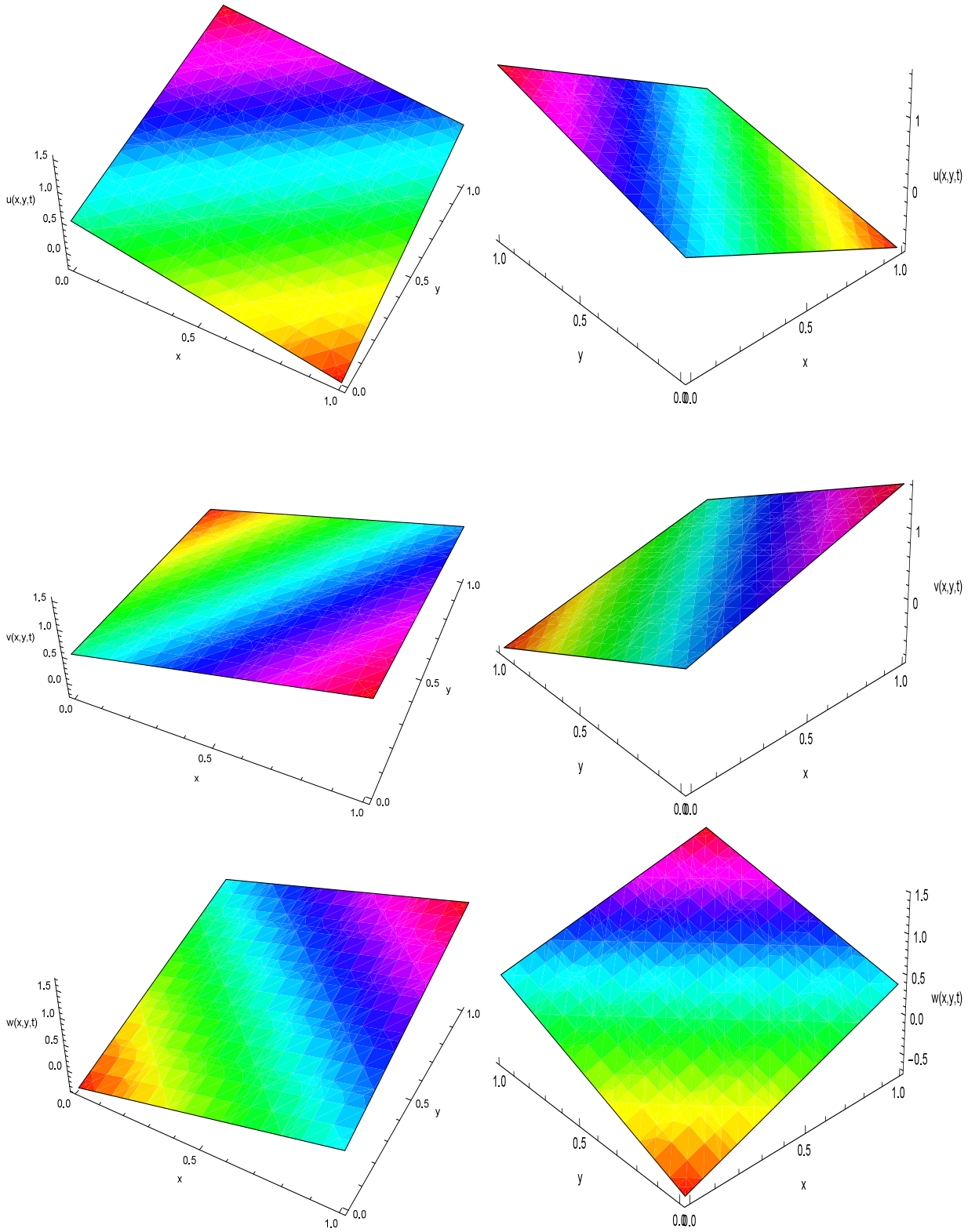


Figure 6.7: The velocity profile $(u, v, w)$ of TFCB equation in Example 6.4.3 with $\alpha = 0.5, 1.0$ (below to above) at $t = 0.1$

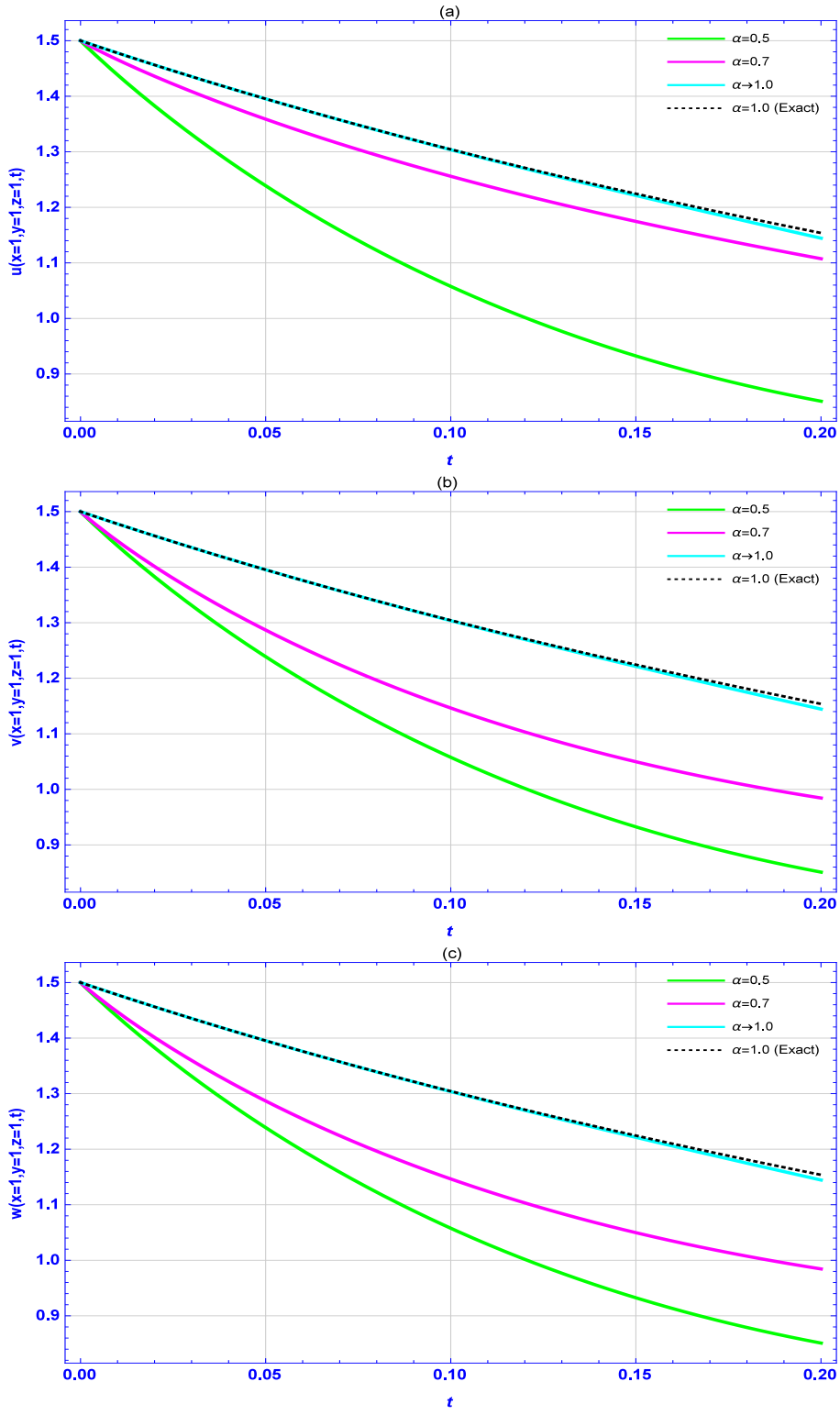


Figure 6.8: The solution behavior of velocity $u, v, w$ of TFCB equation in Example 6.4.3 at $t \leq 0.2$ with $\alpha = 0.5$

*havior of TFCB equation is depicted in Fig. 6.8 for* $\alpha = 0.5, 0.7, 1$*, which shows that the HPM solution profile* $(u, v, w)$*, for* $\alpha = 1$*, converges to exact solutions for* $t < 1$*.*

## 6.5 Conclusion

In this chapter the `homotopy perturbation method` has been adopted for numerical study of $(n+1)$D TFCB equation $(n = 2, 3)$ with appropriate initial conditions, where fractional derivative is of Caputo type. The analytical results are obtained in terms of a power series. Three test problems are carried out in order to validate and illustrate the efficiency of the method. The proposed solutions agreed excellently with exact solutions and FRDTM solutions [248]. The results are also depicted in graphically for different values of the fractional order $\alpha$ and large Reynolds number (Re$\geq$ 100). It is found that the proposed series solutions converges rapidly for large Reynolds numbers. The solutions are approximated without any discretization, transformation or restrictive conditions.

# Chapter 7

# Numerical simulation for time fractional PDEs with proportional delay using two reliable methods

## 7.1 Introduction

Consider an initial value autonomous system of time fractional partial differential equations (TFPDEs) with proportional delay of the form

$$\begin{cases} \mathcal{D}_t^{\alpha}\left(u(x,t)\right) = f\left(x, u(a_0x, b_0t), \frac{\partial}{\partial x}u(a_1x, b_1t), \ldots, \frac{\partial^m}{\partial x^m}u(a_mx, b_mt)\right), \\ u^k(x,0) = \psi_k(x). \end{cases}$$

where $a_i, b_i \in (0,1),\ i \in N \cup \{0\}$. $\psi_k$ is initial value and $f$ is the differential operator, the independent variables $(x,t)$ generally denotes the position in space or size of cells, maturation level at time $t$ while its solution may be the voltage, temperature, densities of different particles, form instance, chemicals, cells, etc..

One significant example of the model: Kortewegde Vries (KdV) equation, arise in the research of shallow water waves is as follow:

$$\mathcal{D}_t^{\alpha}\left(u(x,t)\right) = bu\frac{\partial}{\partial x}u(a_0x, b_0t) + \frac{\partial^3}{\partial x^3}u(a_1x, b_1t), \quad 0 < \alpha < 1.$$

where $b$ is a constant.

The time-fractional nonlinear Klein-Gordon equation with proportional delay, aries in quantum field theory to describe nonlinear wave interaction, is defined as follows

$$\mathcal{D}_t^{\alpha}\left(u(x,t)\right) = u\frac{\partial^2}{\partial x^2}u(a_0x,b_0t) - bu(a_1x,b_1t) - F(u(a_2x,b_2t)) + h(x,t), \quad 1<\alpha<2.$$

where $b$ is a constant, $h(x,t)$ known analytic function and $F$ is the nonlinear operator of $u(x,t)$. For details of various type of models, we refer the reader to [2, 206, 279] and the references therein.

The variational iteration method (VIM) has been developed by Chinese Mathematician He [85]. After the seminal work of He, various modification of VIM has been employed to solve various nonlinear problems, among others, diffusion and wave equations on cantor sets [283], Riccati differential equation [71] and alternative VIM for time fractional Fornberg-Whitham equation [207], for more details, we refer the readers to [188, 189, 207] and the reference therein. The reduced differential transform method (RDTM) has been introduced by Keskin and Oturanc [111] for finding the approximate series solutions of PDEs. After seminal work of Keskin and Oturanc, FRDTM have been adopted to solve vigorous type differential equations arising in mathematics, physics and engineering by various researchers: Srivastava et al. [256, 257], Saravanan and Magesh [211, 212], Yu et al. [292], Singh and Kumar [246], Singh [237] and Singh and Mahendra [251] and many more.

In the past years, very few methods are available for the study of TFPDEs with delay, namely Chebyshev pseudospectral method [299], spectral collocation-waveform relaxation method [300] and iterated pseudospectral method [164], group analysis method [271]. Abazari and Ganji [2] proposed two dimensional differential transform method (2D-DTM) and RDTM for partial differential equations with proportional delay. Abazari and Kilicman [3] adopted DTM for the study of nonlinear integro-differential equations with proportional delay. In [32, 206, 221], the HPM was adopted for solving time fractional partial differential equations with delay. Chen and Wang [44] proposed variational iteration method (VIM) for solving a neutral functional-differential equation with proportional delays. The functional constraints method was proposed by Polyanin and Zhurov [195] for the numerical

study of nonlinear delay reaction-diffusion equations and many more.

The main aim this chapter is to investigate the basic properties of $(n+1)$ dimensional extended fractional reduced differential transform method for time fractional differential partial differential equations with proportional delay. In addition to the above mentioned properties, the initial value autonomous systems of time fractional PDEs with proportional delay has been studied analytical in terms of the series solutions of TFPDEs by employing two reliable methods: alternative variational iteration method *(AVIM)* and $(n+1)$ dimensional extended fractional reduced differential transform method (FRDTM).

## 7.2 Description of the methods

This section deals with the description of variational iteration method and extended fractional reduced differential transform method for time fractional PDEs with proportional delay.

## 7.3 Alternative variational iteration method (AVIM)

Consider a initial value differential equation of the form

$$\begin{aligned} &Lu(t) + Nu(t) = g(t), \\ &u^{(k)}(0) = c_k, \qquad k = 0, 1, \ldots, m-1. \end{aligned} \tag{7.3.1}$$

where $c_k$ are real numbers, $L = \frac{d^m}{dt^m}, m \in \mathbb{N}$ be a linear operator; $N \rightarrow$nonlinear operator and $g(t)$ is a known analytic function.

The correction functional for (7.3.1) can be constructed using AVIM as defined in [189] as:

$$u_{k+1}(t) = u_k(t) + \int_0^t [\lambda(\tau)(Lu_k(\tau) - N\bar{u}_k(\tau) - g(\tau))]d\tau \tag{7.3.2}$$

where the Lagrange multiplier $\lambda(\tau)$ can be identified optimally by means of variation theory.

Generally, the following Lagrange multipliers are used:

$$\lambda(\tau) = \frac{(-1)^m}{(m-1)!}(\tau - t)^{m-1}, \qquad m \geq 1. \tag{7.3.3}$$

Eq. (7.3.3) and (7.3.2) yields the following iteration formula for $(k+1)$th order solution:

$$u_{k+1}(t) = u_k(t) + A[u_k(t)], \tag{7.3.4}$$

where the operator $A[u]$ is defined by

$$A(u) := \frac{(-1)^m}{(m-1)!}\int_0^t \left((\tau - t)^{m-1}(Lu_k(\tau) - N\bar{u}_k(\tau) - g(\tau))\right) d\tau \tag{7.3.5}$$

Moreover, if we set the components $s_k$ $(k = 0, 1, 2\ldots)$ as:

$$\begin{cases} s_0 = u_0 \\ s_1 = A[s_0] \\ s_2 = A[s_0 + s_1] \\ \vdots \\ s_{k+1} = A[s_0 + s_1 + \ldots + s_k] \end{cases} \tag{7.3.6}$$

From (7.3.6), we get $s_0+s_1 = u_0+A[u_0] = u_1$; $s_0+s_1+s_2 = u_1+A[u_1] = u_2$; $s_0+s_1+s_2+s_3 = u_1 + A[u_1] + A[s_0 + s_1 + s_2] = u_2 + A[u_2] = u_3$ and consequently, the approximate solution of (7.3.1) is given by

$$u(t) = \lim_{k\to\infty} u_k(t) = \sum_{k=0}^{\infty} s_k(t). \tag{7.3.7}$$

The interested readers are referred to [71, 188, 189, 207, 283] for further detail.

### 7.3.1 AVIM for time fractional PDEs

Consider the initial value system of time fractional PDE of the form

$$\begin{cases} \mathcal{D}_t^{\alpha}\{u(x,t)\} + L[u(x,t)] + N[u(x,t)] = g(x,t),\ m-1 < \alpha \leq m \in \mathbb{N} \\ u^{(k)}(x,0) = f_k(x),\ k = 0,1,\ldots,m-1, x \in R, \end{cases} \tag{7.3.8}$$

where $L$ and $N$ are linear and nonlinear operators respectively, $g = g(x,t)$ is the known analytic function, $\mathcal{D}_t^{\alpha}$ is the Caputo fractional derivative of order $\alpha$ and $f_k's$ is a real valued function.

The solution

$$u(x,t) = \lim_{n\to\infty} u_n(x,t)$$

to the problem (7.3.8) can be derived from the following variation iteration formula as in [188]:

$$u_{k+1}(t) = u_k(t) + J_t^{\alpha}[\lambda(t,\tau)\left(\mathcal{D}_t^{\alpha}u_k(x,t) + L[u_k(x,t)] + N[u_k(x,t)] - g(x,t)\right)] \tag{7.3.9}$$

the terms $L[u_k(x,t)]$ and $N[u_k(x,t)]$ are restricted variations, and the Lagrange multiplier can be computed as $\lambda(t,\tau) = -1$.

The variational iteration solution: $u(x,t) = \sum_{k=0}^{\infty} v_k(x,t)$, in the present framework is obtained by the following iteration formula for $\lceil\alpha\rceil = m \in \mathbb{N}$ as in [188, 189]:

$$\begin{cases} v_0 = \displaystyle\sum_{k=0}^{n-1} \frac{f_k(x)}{k!}t^k \\ v_{k+1} = -\dfrac{\Gamma(\alpha)}{\Gamma(\alpha-m+1)\Gamma(m)}\times \\ \qquad \mathcal{J}_t^{\alpha}\left[\mathcal{D}_t^{\alpha}\left[v_0+\ldots+v_k\right] + L\left[v_0+\ldots+v_k\right] + N\left[v_0+\ldots+v_k\right] - g(x,t)\right], \end{cases} \tag{7.3.10}$$

The iteration formula (7.3.10) converges to a solution of problem (7.3.8) whenever there exists $\gamma$ such that $\gamma \in (0,1)$ and $s_k \leq \gamma s_{k-1}\ \forall\ k \in \mathbb{N}$.

**Theorem 7.3.1** (Convergence analysis [188,189]). *Let $A$ be an operator as defined in* (7.3.5) *from Banach space to Banach space, then*

*a) the series solution* $u(x,t) = \sum_{k=0}^{\infty} s_k(x,t)$ *as defined in* (7.3.7) *converges whenever* $\exists(0 < \sigma < 1)$ *such that* $A[s_0 + s_1 + \ldots + s_{k+1}] \leq \sigma A[s_0 + s_1 + \ldots + s_k]$ *(i.e.,* $s_{k+1} \leq \sigma s_k$*)* $\forall\ k \in \mathbb{N} \cup \{0\}$*, and*

*b) the maximum absolute truncation error in solution* (7.3.7) *for problem* (7.3.8) *is computed as*

$$\left|\left| u(x,t) - \sum_{i=0}^{\ell} s_i(x,t) \right|\right| \leq \frac{\sigma^{\ell+1}}{1-\sigma} ||s_0(x,t)||. \tag{7.3.11}$$

The complete proof of the above results is given in [188, 189].

## 7.4 Extended fractional reduced differential transform method

The properties of FRDTM has been extended for $(n+1)$ dimensional time fractional partial differential equations with proportional delay in the following

### 7.4.1 $(n+1)$-dimensional extended FRDTM

Let $X = (x_1, x_2, \ldots, x_n)$ be a vector of $n$ variables, and let $X_a = (a_1x_1, a_2x_2, \ldots, a_nx_n)$, where $a_i \in (0,1)$.

**Theorem 7.4.1.** *Let* $\Psi_\alpha^k(X)$ *and* $\Phi_\alpha^k(X)$ *be the spectrums of the analytic and continuously differentiable functions* $\psi(X,t)$ *and* $\phi(X,t)$*, respectively. Then*

(1) *If* $\theta(X,t) = \ell\psi(X_a, bt)$*. Then*

$$\Theta_\alpha^k(X) = \ell b^{\alpha k} \Psi_\alpha^k(X_a).$$

(2) *If* $\theta(X,t) = \ell_1\psi(X_a, b_1t) \pm \ell_2\phi(X_a, b_2t)$*. Then*

$$\Theta_\alpha^k(X) = \ell_1 b_1^{k\alpha} \Psi_\alpha^k(X_a) \pm \ell_2 b_2^{k\alpha} \Phi_\alpha^k(X_a).$$

(3) *If* $\theta(X,t)=\psi\left(X_a,b_1t\right)\phi\left(X_c,b_2t\right)$. *Then*

$$\Theta_\alpha^k(X)=\sum_{r=0}^{k}b_1^{r\alpha}b_2^{(k-r)\alpha}\Psi_\alpha^r(X_a)\Phi_\alpha^{k-r}(X_c).$$

(4) *If* $\theta(x,t)=f(X)\psi(X_a,bt)$. *Then*

$$\Theta_\alpha^k(X)=f(X)b^{k\alpha}\Psi_\alpha^k\left(X_a\right).$$

(5) *If* $\theta(X,t)=\frac{\partial^{r_i+n\alpha}}{\partial x_i^{r_i}\partial t^{n\alpha}}\left\{\psi\left(X_a,bt\right)\right\}$. *Then*

$$\Theta_\alpha^k(X)=b^{(k+n)\alpha}\frac{\Gamma(1+(k+n)\alpha)}{\Gamma(1+k\alpha)}\frac{\partial^{r_i}}{\partial x_i^{r_i}}\left\{\Psi_\alpha^{k+n}\left(X_a\right)\right\}$$

*In particular,*

(a) *If* $\theta(X,t)=\frac{\partial^{r_i}}{\partial x_i^{r_i}}\left\{\psi\left(X_a,bt\right)\right\}$. *Then*

$$\Theta_\alpha^k(X)=b^{k\alpha}\frac{\partial^{r_i}}{\partial x_i^{r_i}}\left\{\Psi_\alpha^k\left(X_a\right)\right\}$$

(b) *If* $\theta(X,t)=\frac{\partial^{n\alpha}}{\partial t^{n\alpha}}\left\{\psi\left(X_a,bt\right)\right\}$. *Then*

$$\Theta_\alpha^k(X)=b^{(k+n)\alpha}\frac{\Gamma(1+(k+n)\alpha)}{\Gamma(1+k\alpha)}\left\{\Psi_\alpha^{k+n}\left(X_a\right)\right\}$$

*Proof.* (1) Let $\tau=bt$, then $b\partial t=\partial\tau$. Thus,

$$\begin{aligned}\Theta_\alpha^k(X)&=\frac{1}{\Gamma(1+k\alpha)}\left\{\frac{\partial^{k\alpha}}{\partial t^{k\alpha}}\theta(X,t)\right\}_{t=0}=\frac{b^{k\alpha}}{\Gamma(1+k\alpha)}\left\{\frac{\partial^{k\alpha}}{\partial\tau^{k\alpha}}\ell\psi(X_a,\tau)\right\}_{\tau=0}\\&=\ell b^{k\alpha}\left[\frac{1}{\Gamma(1+k\alpha)}\left\{\frac{\partial^{k\alpha}}{\partial\tau^{k\alpha}}\psi(X_a,\tau)\right\}_{\tau=0}\right]=\ell b^{k\alpha}\Psi_\alpha^k(X_a).\end{aligned}\tag{7.4.1}$$

(2) Keeping $\frac{\partial^{k\alpha}}{\partial t^{k\alpha}}\left\{\theta(X,t)\right\}=\ell_1\frac{\partial^{k\alpha}}{\partial t^{k\alpha}}\left\{\psi\left(X_a,b_1t\right)\right\}\pm\ell_2\frac{\partial^{k\alpha}}{\partial t^{k\alpha}}\left\{\phi\left(X_a,b_2t\right)\right\}$ with property (1), we get

$$\Theta_\alpha^k(X)=\ell_1b_1^{k\alpha}\Psi_\alpha^k\left(X_a\right)\pm\ell_2b_2^{k\alpha}\Phi_\alpha^k\left(X_a\right).$$

(3) Set $u(X,t) = \psi(X_a, b_1 t)$ and $v(X,t) = \phi(X_c, b_2 t)$, then Theorem 7.4.1(1) confirms that $U_\alpha^r(X) = b_1^{r\alpha}\Psi_\alpha^r(X_a)$ and $V_\alpha^{k-r}(X) = b_2^{(k-r)\alpha}\Phi_\alpha^{k-r}(X_a)$. Using Theorem 1.3.1($a$) for $\theta(X,t) = u(X,t)v(X,t)$, we get

$$\begin{aligned}\Theta_\alpha^k(X) &= \sum_{r=0}^{k} U_\alpha^r(X)V_\alpha^{k-r}(X) = \sum_{r=0}^{k}\left\{b_1^{r\alpha}\Psi_\alpha^r(X_a)\right\}\left\{b_2^{(k-r)\alpha}\Phi_\alpha^{k-r}(X_c)\right\} \\ &= \sum_{r=0}^{k} b_1^{r\alpha}\ b_2^{(k-r)\alpha}\Psi_\alpha^r(X_a)\Phi_\alpha^{k-r}(X_c).\end{aligned} \tag{7.4.2}$$

(4) Let $\tau = bt$, then $b\partial t = \partial\tau$ and let $\theta(x,t) = f(X)\psi(X_a, bt)$. Then

$$\begin{aligned}\Theta_\alpha^k(X) &= \frac{1}{\Gamma(1+k\alpha)}\left\{\frac{\partial^{k\alpha}}{\partial t^{k\alpha}}\theta(X_a, bt)\right\}_{t=0} \\ &= \frac{f(X)b^{k\alpha}}{\Gamma(1+k\alpha)}\left\{\frac{\partial^{k\alpha}}{\partial \tau^{k\alpha}}\psi(X_a,\tau)\right\}_{\tau=0} = b^{k\alpha}f(X)\Psi_\alpha^k(X_a).\end{aligned} \tag{7.4.3}$$

(5) Let $\theta(X,t) = \frac{\partial^{r_i+n\alpha}}{\partial x_i^{r_i}\partial t^{n\alpha}}\left\{\psi(X_a, bt)\right\}$. Using Theorem 7.4.1(1), we get

$$\begin{aligned}\Theta_\alpha^k(X) &= \frac{1}{\Gamma(1+k\alpha)}\left[\frac{\partial^{k\alpha}}{\partial t^{k\alpha}}\left\{\theta(X,t)\right\}\right]_{t=0} \\ &= \frac{b^{(k+n)\alpha}}{\Gamma(1+k\alpha)}\frac{\partial^{r_i}}{\partial x_i^{r_i}}\left[\frac{\partial^{(k+n)\alpha}}{\partial \tau^{(k+n)\alpha}}\left\{\psi(X_a,\tau)\right\}\right]_{\tau=0} \\ &= b^{(k+n)\alpha}\frac{\Gamma(1+(k+n)\alpha)}{\Gamma(1+k\alpha)}\frac{\partial^{r_i}}{\partial x_i^{r_i}}\left\{\Psi_\alpha^{k+n}(X_a)\right\}\end{aligned} \tag{7.4.4}$$

Moreover, for $n = 0$, we get result 5(a), and for $r_i = 0$ we get result 5(b).

■

### 7.4.2 Extended FRDTM for TFPDEs with proportional delay

Consider the initial value autonomous system of TFPDEs with proportional delay as follows:

$$\begin{cases}\mathcal{D}_t^\alpha\left(u(x,t)\right) = f\left(x, u(a_0x, b_0t), \frac{\partial}{\partial x}u(a_1x, b_1t), \ldots, \frac{\partial^m}{\partial x^m}u(a_mx, b_mt)\right), \\ u(x,0) = \psi(x).\end{cases} \tag{7.4.5}$$

The extended FRDTM on Equation (7.4.5) yields a recurrence relation in $U_\alpha^k$'s of the following form

$$\begin{cases} \dfrac{\Gamma(1+(1+k)\alpha)}{\Gamma(1+k\alpha)} U_\alpha^{k+1}(x) = F\left(x, b_0^\alpha, \ldots, b_m^\alpha, U_\alpha^0(a_0x), \ldots, U_\alpha^k(a_0x), \dfrac{\partial U_\alpha^0(a_1x)}{\partial x}, \right. \\ \qquad\qquad \left. \ldots, \dfrac{\partial U_\alpha^k(a_1x)}{\partial x}, \ldots, \dfrac{\partial^m U_\alpha^0(a_mx)}{\partial x^m}, \ldots, \dfrac{\partial^m U_\alpha^k(a_mx)}{\partial x^m}\right) \\ U_\alpha^0(x) = \psi(x), \end{cases} \tag{7.4.6}$$

which is a recurrence relation in $U_\alpha^k$, $k = 0, 1, 2, \cdots$. This relation can be solved for $U_\alpha^k$ for each $k$ once $\psi(x)$ is known. The inverse FRDTM leads the approximate series solution $u(x,t)$ of TFPDEs with proportional delay.

## 7.5 Results and discussion

This section deals with the main goal of this chapter, a new application of alternative variational iteration method and extended FRDTM in computation of the approximate solutions of the initial value autonomous system of TFPDEs with proportional delay. The effectiveness and validity of both of the methods is illustrated by considering three test problems.

**Example 7.5.1.** *Consider initial value system of time-fractional order, generalized Burgers' equation with proportional delay as given in [2, 206]*

$$\begin{cases} \mathcal{D}_t^\alpha u(x,t) = \dfrac{\partial^2}{\partial x^2} u(x,t) + u\left(\dfrac{x}{2}, \dfrac{t}{2}\right) \dfrac{\partial}{\partial x} u\left(x, \dfrac{t}{2}\right) + \dfrac{1}{2} u(x,t) \\ u(x,0) = x, \end{cases} \tag{7.5.1}$$

*In particular for $\alpha = 1$ the exact solution is $u(x,t) = x\exp(t)$.*

**Example 7.5.2.** *Consider initial value TFPDE with proportional delay as given in [206]*

$$\begin{cases} \mathcal{D}_t^\alpha u(x,t) = u\left(x, \dfrac{t}{2}\right) \dfrac{\partial^2}{\partial x^2} u\left(x, \dfrac{t}{2}\right) - u(x,t) \\ u(x,0) = x^2, \end{cases} \tag{7.5.2}$$

*In particular for $\alpha = 1$ the exact solution is $u(x,t) = x^2 \exp(t)$.*

**Example 7.5.3.** *Consider initial value TFPDE with proportional delay as given in [2, 206]*

$$\begin{cases} \mathcal{D}_t^{\alpha} u(x,t) = \dfrac{\partial^2}{\partial x^2} u\left(\dfrac{x}{2}, \dfrac{t}{2}\right) \dfrac{\partial}{\partial x} u\left(\dfrac{x}{2}, \dfrac{t}{2}\right) - \dfrac{1}{8} \dfrac{\partial}{\partial x} u\,(x,t) - u(x,t) \\ u(x,0) = x^2, \end{cases} \tag{7.5.3}$$

*In particular, for $\alpha = 1$, the exact solution is $u(x,t) = x^2 \exp(-t)$.*

### 7.5.1 Solution of Example 7.5.1

#### A. AVIM

Keeping (7.3.10) in mind, the iteration formula for (7.5.1) can be constructed as

$$\begin{cases} s_0 = x \\ s_{k+1} = -\mathcal{J}_t^{\alpha} \left[ \mathcal{D}_t^{\alpha} \displaystyle\sum_{i=0}^{k} s_i(x,t) - \sum_{i=0}^{k} \left( \frac{\partial^2 s_i(x,t)}{\partial x^2} - \frac{1}{2} s_i(x,t) \right) + \sum_{i=0}^{k} s_i \left( \frac{x}{2}, \frac{t}{2} \right) \frac{\partial s_{k-i}(x, \frac{t}{2})}{\partial x} \right] \end{cases}$$

On simplifying the above relation, we get

$$\begin{cases} s_0 = x, \\ s_1 = \dfrac{x t^{\alpha}}{\Gamma(\alpha+1)}, \\ s_2 = \dfrac{x(2^{1-\alpha}+1) t^{2\alpha}}{2\Gamma(2\alpha+1)}, \\ s_3 = 2^{-2-3\alpha} x \left[ \dfrac{(2+2^{\alpha})(2+4^{\alpha})(\Gamma(1+\alpha))^2 + 2^{1+\alpha}\Gamma(1+2\alpha) t^{3\alpha}}{(\Gamma(1+\alpha))^2 \Gamma(1+3\alpha)} \right], \\ \vdots \end{cases} \tag{7.5.4}$$

The approximate solution of the problem (7.5.1) is

$$\begin{aligned} u(x,t) &= s_0(x,t) + s_1(x,t) + s_2(x,t) + s_3(x,t) + \ldots \\ &= x \left\{ 1 + \frac{t^{\alpha}}{\Gamma(\alpha+1)} + \frac{(2^{1-\alpha}+1) t^{2\alpha}}{2\Gamma(2\alpha+1)} \right. \\ &\qquad \left. + \left( \frac{(2+2^{\alpha})(2+4^{\alpha})}{2^{2+3\alpha}\Gamma(1+3\alpha)} + \frac{2^{-1-2\alpha}\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2 \Gamma(1+3\alpha)} \right) t^{3\alpha} + \ldots \right\} \end{aligned} \tag{7.5.5}$$

**B. Extended FRDTM**

Using extended FRDTM on (7.5.1), the recurrence relation in $U_\alpha^k(x)$ is obtained as

$$\begin{cases} \dfrac{\Gamma\left(1+(1+k)\alpha\right)}{\Gamma\left(1+k\alpha\right)} U_\alpha^{k+1}(x) = \dfrac{\partial^2}{\partial x^2} U_\alpha^k(x) + \dfrac{1}{2^{k\alpha}} \displaystyle\sum_{r=0}^{k} U_\alpha^{k-r} \left(\frac{x}{2}\right) \frac{\partial}{\partial x} U_\alpha^r\left(x\right) + \frac{1}{2} U_\alpha^k(x) \\ U_\alpha^0(x) = x, \end{cases} \tag{7.5.6}$$

The solution of recurrence relation (7.5.6) is obtained as follows

$$\begin{aligned} U_\alpha^0(x) &= x, \\ U_\alpha^1(x) &= \frac{x}{\Gamma(1+\alpha)} \\ U_\alpha^2(x) &= \frac{(1+2^{1-\alpha})x}{2\Gamma(1+2\alpha)} \equiv Ax \\ U_\alpha^3(x) &= \frac{x}{4\Gamma(1+3\alpha)} \left\{ 1 + 2^{1-\alpha} + 2^{1-2\alpha} + 2^{1-3\alpha} + \frac{\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}\ 2^{1-2\alpha} \right\} \equiv Bx \\ U_\alpha^4(x) &= \frac{\Gamma(1+3\alpha)}{2\Gamma(1+4\alpha)} \left( (1+2^{1-3\alpha})B + \frac{(1+2^{1-3\alpha})}{(1+2^{1+\alpha})} A \right) x \\ &\vdots \qquad\qquad\qquad \vdots \end{aligned} \tag{7.5.7}$$

Using inverse FRDTM, we get

$$\begin{aligned} u(x,t) &= U_\alpha^0(x) + U_\alpha^1(x)t^\alpha + U_\alpha^2(x)t^{2\alpha} + U_\alpha^3(x)t^{3\alpha} + U_\alpha^4(x)t^{4\alpha} + \dots \\ &= x \left( 1 + \frac{t^\alpha}{\Gamma(1+\alpha)} + \frac{(1+2^{1-\alpha})}{2\Gamma(1+2\alpha)} t^{2\alpha} + Bt^{3\alpha} \right. \\ &\quad \left. + \frac{\Gamma(1+3\alpha)}{2\Gamma(1+4\alpha)} \left( (1+2^{1-3\alpha})B + \frac{(1+2^{1-3\alpha})}{(1+2^{1+\alpha})} A \right) t^{4\alpha} + \dots \right) \end{aligned} \tag{7.5.8}$$

**Remark 7.5.4.** *The computed solutions obtained by both of the methods is a closed form to the results obtained by Sarkar et al [206]. The computed AVIM solutions $u(x,t)$ are depicted in Figure 7.1(a-c) and two dimensional plots for different values of $\alpha = 0.8, 0.9, 1.0$ at different time levels $t \leq 1$ with $x = 1$ are depicted in Figure 7.2(a). In particular for $\alpha = 1$, both the AVIM olution* (7.5.5) *and FRDTM solutions* (7.5.8) *reduces to*

$$u(x,t) = x \left( 1 + t + \frac{t^2}{2!} + \frac{t^3}{3!} + \frac{t^4}{4!} + \dots \right) \tag{7.5.9}$$

Table 7.1: Approximate AVIM solution of Example 7.5.1 for $\alpha = 1$

| $x$ | $t$ | Exact | AVIM | | | |
|---|---|---|---|---|---|---|
| | | | Approx. | $\|u - u_3\|$ | $\|u - u_5\|$ | $\|u - u_7\|$ |
| 0.25 | 0.25 | 3.210064E-01 | 3.210063E-01 | 4.281251E-05 | 8.789589E-08 | 9.730583E-11 |
| | 0.50 | 4.121803E-01 | 4.121745E-01 | 7.219843E-04 | 5.838508E-06 | 2.563634E-08 |
| | 0.75 | 5.292500E-01 | 5.291809E-01 | 3.859379E-03 | 6.909595E-05 | 6.766298E-07 |
| | 1.00 | 6.795705E-01 | 6.791667E-01 | 1.290379E-02 | 4.037904E-04 | 6.965051E-06 |
| 0.5 | 0.25 | 6.420127E-01 | 6.420125E-01 | 8.562501E-05 | 1.757918E-07 | 1.946117E-10 |
| | 0.50 | 8.243606E-01 | 8.243490E-01 | 1.443969E-03 | 1.167702E-05 | 5.127268E-08 |
| | 0.75 | 1.058500E+0 | 1.058362E+0 | 7.718758E-03 | 1.381919E-04 | 1.353260E-06 |
| | 1.00 | 1.359141E+0 | 1.358333E+0 | 2.580758E-02 | 8.075809E-04 | 1.393010E-05 |
| 0.75 | 0.25 | 9.630191E-01 | 9.630188E-01 | 1.284375E-04 | 2.636877E-07 | 2.919175E-10 |
| | 0.50 | 1.236541E+0 | 1.236523E+0 | 2.165953E-03 | 1.751553E-05 | 7.690902E-08 |
| | 0.75 | 1.587750E+0 | 1.587543E+0 | 1.157814E-02 | 2.072879E-04 | 2.029889E-06 |
| | 1.00 | 2.038711E+0 | 2.037500E+0 | 3.871137E-02 | 1.211371E-03 | 2.089515E-05 |

Table 7.2: Extended FRDTM solutions with first five terms of Example 7.5.1 for $\alpha = 0.8, 0.9$ and their comparisons with [206] for $\alpha = 1$

| $x$ | $t$ | Exact | HPM [206] | | Extended FRDTM | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Approx. | $E_{abs}$ | Approx.($\alpha = 1$) | $E_{abs}(\alpha = 1)$ | $\alpha = 0.8$ | $\alpha = 0.9$ |
| 0.25 | 0.25 | 3.210064E-01 | 3.210042E-01 | 2.123000E-06 | 3.210064E-01 | 3.952908E-09 | 3.630319E-01 | 3.390673E-01 |
| | 0.50 | 4.121803E-01 | 4.121094E-01 | 7.094300E-05 | 4.121799E-01 | 4.134861E-07 | 4.900740E-01 | 4.455736E-01 |
| | 0.75 | 5.292500E-01 | 5.286865E-01 | 5.634830E-04 | 5.292427E-01 | 7.298701E-06 | 6.524126E-01 | 5.815108E-01 |
| | 1.00 | 6.795705E-01 | 6.770833E-01 | 2.487123E-03 | 6.795139E-01 | 5.656911E-05 | 8.628138E-01 | 7.565018E-01 |
| 0.50 | 0.25 | 6.420127E-01 | 6.420085E-01 | 4.245000E-06 | 6.420127E-01 | 7.905816E-09 | 7.260638E-01 | 6.781345E-01 |
| | 0.50 | 8.243606E-01 | 8.242188E-01 | 1.418850E-04 | 8.243598E-01 | 8.259722E-07 | 9.801479E-01 | 8.911471E-01 |
| | 0.75 | 1.058500E+0 | 1.057373E+0 | 1.126970E-03 | 1.058485E+0 | 1.459740E-05 | 1.304825E+0 | 1.163022E+0 |
| | 1.00 | 1.359141E+0 | 1.354167E+0 | 4.974250E-03 | 1.359028E+0 | 1.131422E-04 | 1.725628E+0 | 1.513004E+0 |
| 0.75 | 0.25 | 9.630191E-01 | 9.630127E-01 | 6.367000E-06 | 9.630191E-01 | 1.185872E-08 | 1.089096E+0 | 1.017202E+0 |
| | 0.50 | 1.236541E+0 | 1.236328E+0 | 2.128300E-04 | 1.236540E+0 | 1.236458E-06 | 1.470222E+0 | 1.336721E+0 |
| | 0.75 | 1.587750E+0 | 1.586060E+0 | 1.690450E-03 | 1.587728E+0 | 2.190110E-05 | 1.957238E+0 | 1.744532E+0 |
| | 1.00 | 2.038711E+0 | 2.031250E+0 | 7.461370E-03 | 2.038542E+0 | 1.697033E-04 | 2.588441E+0 | 2.269505E+0 |

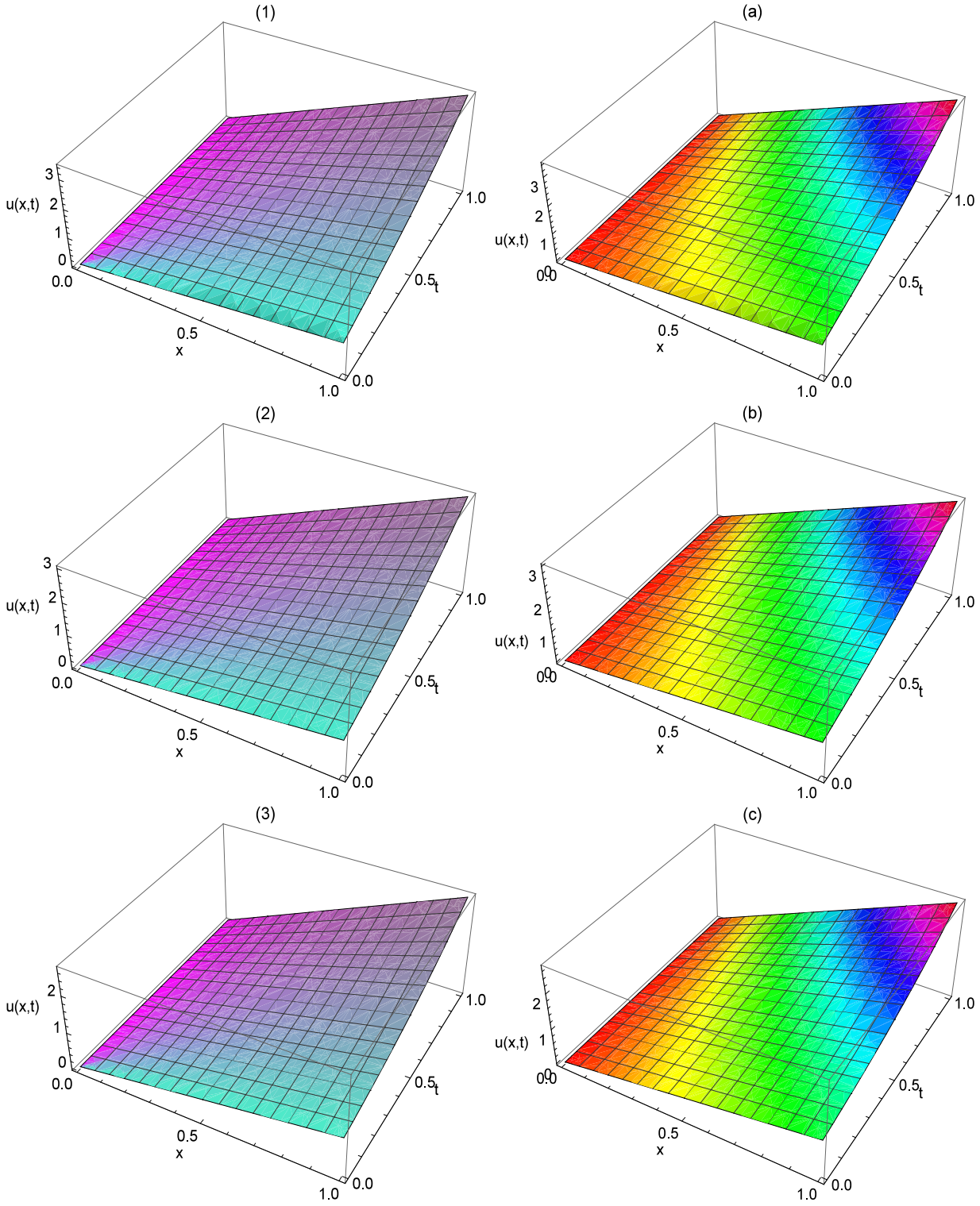


Figure 7.1: The solution behavior of AVIM solutions (a) to (c) and extended FRDTM solutions (1) to (3) of Example 7.5.1 for $\alpha = 0.8; 0.9; 1.0$ (top to bottom)

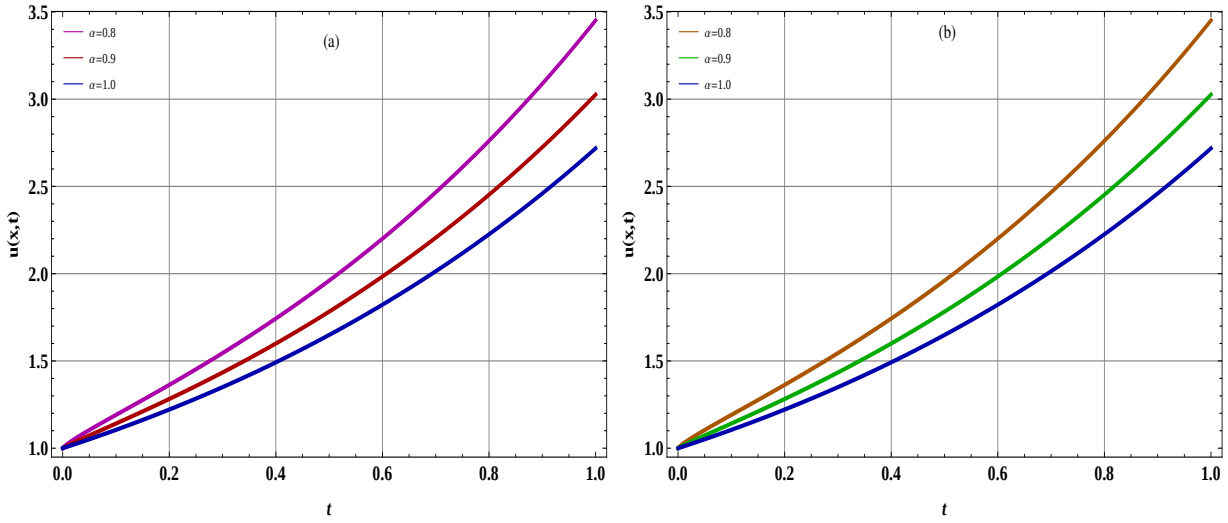


Figure 7.2: Two dimensional plots of (a) AVIM solutions, (b) extended FRDTM solutions of Example 7.5.1 for different values of $\alpha = 0.8, 0.9, 1.0$ at $t \in (0, 1)$ and $x = 1$

*The solution is same as the solution due to DTM and RDTM [2], converges to the exact solution* $u(x,t) = x\exp(t)$.

*For* $\alpha = 1$*, the computed AVIM solutions with first five terms are compared with exact solutions in Table 7.1 and the absolute errors are also computed for different order* ($m = 3, 5, 7$) *solutions is also reported in Table 7.1, which shows that the absolute errors are decreasing with increasing the order of the approximate solutions. Thus, computed results converges to the exact results. The comparison of FRDTM solution with HPM solution [206] for* $\alpha = 1$ *and the extended FRDTM solutions for* $\alpha = 0.8, 0.9$ *are reported in Table 7.2. The computed extended FRDTM solutions* $u(x,t)$ *are depicted in Figure 7.1(1-3) for different values of* $\alpha = 0.8, 0.9, 1.0$ *and Figure 7.2(1) depicts the* $2D$ *solutions for* $\alpha = 0.8, 0.9, 1.0$ *and* $x = 1$ *at different time levels* $t \leq 1$*. The findings show that the computed AVIM solutions and extended FRDTM solutions are agreed well with the solutions obtained HPM [206], DTM and RDTM [2](for* $\alpha = 1$*) and approaching to the exact solutions.*

### 7.5.2 Solution of Example 7.5.2

**A. AVIM**

The iteration formula for (7.5.2) can be constructed as

$$\begin{cases} s_0 = x^2 \\ s_{k+1} = -\mathcal{J}_t^{\alpha}\left[\mathcal{D}_t^{\alpha}\left\{\sum_{i=0}^{k} s_i(x,t)\right\} + \sum_{i=0}^{k} s_i(x,t) - \sum_{i=0}^{k} s_i\left(x,\frac{t}{2}\right)\frac{\partial^2 s_{k-i}\left(x,\frac{t}{2}\right)}{\partial x^2}\right] \end{cases} \tag{7.5.10}$$

On solving the above relation, we get

$$\begin{cases} s_0 = x^2, \\ s_1 = \dfrac{t^{\alpha}x^2}{\Gamma(\alpha+1)}, \\ s_2 = \dfrac{t^{2\alpha}x^2(-2^{-\alpha}(-4+2^{\alpha}))}{\Gamma(2\alpha+1)}, \\ s_3 = 8^{-\alpha}t^{3\alpha}x^2\left[\dfrac{(-4+2^{\alpha})(-2+2^{\alpha})(2+2^{\alpha})\Gamma(1+\alpha)^2 + 2^{1+\alpha}\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2\Gamma(1+3\alpha)}\right], \\ \vdots \end{cases} \tag{7.5.11}$$

The approximate AVIM solution of the problem (7.5.2) is given by

$$\begin{aligned} u(x,t) &= s_0(x,t) + s_1(x,t) + s_2(x,t) + s_3(x,t) + \dots \\ &= x^2\left[1 + \frac{t^{\alpha}}{\Gamma(\alpha+1)} + \frac{t^{2\alpha}(-2^{-\alpha}(-4+2^{\alpha}))}{\Gamma(2\alpha+1)}\right. \\ &\quad \left. + 8^{-\alpha}t^{3\alpha}\left[\frac{(-4+2^{\alpha})(-2+2^{\alpha})(2+2^{\alpha})\Gamma(1+\alpha)^2 + 2^{1+\alpha}\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2\Gamma(1+3\alpha)}\right] + \dots\right] \end{aligned} \tag{7.5.12}$$

which is closed form to the exact solution and the results due to Sarkar et al [206].

**B. Extended FRDTM**

Using extended FRDTM on (7.5.2), the recurrence relation in $U_\alpha^k(x)$ is obtained as follows:

$$\begin{cases} \dfrac{\Gamma\left(1+(1+k)\alpha\right)}{\Gamma\left(1+k\alpha\right)} U_\alpha^{k+1}(x) = \dfrac{1}{2^{k\alpha}} \displaystyle\sum_{r=0}^{k} U_\alpha^{k-r}\left(x\right) \frac{\partial^2}{\partial x^2} U_\alpha^{r}\left(x\right) - U_\alpha^k(x) \\ U_\alpha^0(x) = x^2, \end{cases} \tag{7.5.13}$$

The solution of the above recurrence relation yields the following

$$\begin{aligned} U_\alpha^0(x) &= x^2, \\ U_\alpha^1(x) &= \frac{x^2}{\Gamma(1+\alpha)}, \\ U_\alpha^2(x) &= \frac{(2^{2-\alpha}-1))x^2}{\Gamma(1+2\alpha)}, \\ U_\alpha^3(x) &= \left\{ \frac{2^{4-3\alpha}-4^{-\alpha}-2^{2-\alpha}+1}{\Gamma(1+3\alpha)} + \frac{2^{1-2\alpha}\Gamma(1+2\alpha)}{\Gamma(1+3\alpha)\Gamma(1+\alpha)^2} \right\} x^2 \\ U_\alpha^4(x) &= -\frac{2^{-6\alpha}x^2}{\Gamma(1+4\alpha)} \left\{ \frac{(2^\alpha-4)2^{2+2\alpha}\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)} + \right. \\ &\quad \left. \left( \frac{2^{1+\alpha}(8^\alpha-4)\Gamma(1+2\alpha)}{\left(\Gamma(1+\alpha)\right)^2} - 64 + 2^{4+\alpha} + 3\ 2^{2+3\alpha} - 2^{2+5\alpha} + 2^{4+2\alpha} - 2^{2+4\alpha} + 2^{6\alpha} \right) \right\} \\ &\quad \vdots \qquad\qquad \vdots \end{aligned}$$

On taking inverse FRDTM, we get

$$\begin{aligned} u(x,t) = x^2 &\left( 1 + \frac{t^\alpha}{\Gamma(1+\alpha)} + \frac{(2^{2-\alpha}-1)}{\Gamma(1+2\alpha)} t^{2\alpha} + \left\{ \frac{2^{4-3\alpha}-4^{-\alpha}-2^{2-\alpha}+1}{\Gamma(1+3\alpha)} \right\} t^{3\alpha} \right. \\ &+ \left\{ \frac{2^{1-2\alpha}\Gamma(1+2\alpha)}{\Gamma(1+3\alpha)\Gamma(1+\alpha)^2} \right\} t^{3\alpha} - \frac{64^{-\alpha}x^2}{\Gamma(1+4\alpha)} \left\{ \frac{(2^\alpha-4)4^{1+\alpha}\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)} \right\} t^{4\alpha} - \\ &\frac{64^{-\alpha}x^2}{\Gamma(1+4\alpha)} \left\{ \left( \frac{2^{1+\alpha}(8^\alpha-4)\Gamma(1+2\alpha)}{\left(\Gamma(1+\alpha)\right)^2} \right. \right. \\ &\left. \left. -64 + 2^{4+\alpha} + 3\ 2^{2+3\alpha} - 2^{2+5\alpha} + 4^{2+\alpha} - 4^{1+2\alpha} + 64^\alpha \right) \right\} t^{4\alpha} + \ldots \end{aligned} \tag{7.5.14}$$

**Remark 7.5.5.** *The fifth order AVIM solutions for $\alpha = 1$ are compared with exact solutions and the absolute errors in different orders ($m = 3, 5, 7$) solutions are reported in Table 7.3.*

Table 7.3: Approximate AVIM solutions of Example 7.5.2 for $\alpha = 1$

| $x$ | $t$ | Exact | AVIM | | | |
|---|---|---|---|---|---|---|
| | | | Approx. | $\|u - u_3\|$ | $\|u - u_5\|$ | $\|u - u_7\|$ |
| 0.25 | 0.25 | 8.025159E-02 | 8.025157E-02 | 1.070313E-05 | 2.197397E-08 | 1.489440E-09 |
| | 0.50 | 1.030451E-01 | 1.030436E-01 | 1.804961E-04 | 1.459627E-06 | 1.873533E-07 |
| | 0.75 | 1.323125E-01 | 1.322952E-01 | 9.648448E-04 | 1.727399E-05 | 3.141455E-06 |
| | 1.00 | 1.698926E-01 | 1.697917E-01 | 3.225948E-03 | 1.009476E-04 | 2.306032E-05 |
| 0.50 | 0.25 | 3.210064E-01 | 3.210063E-01 | 4.281251E-05 | 8.789589E-08 | 5.957769E-09 |
| | 0.50 | 4.121803E-01 | 4.121745E-01 | 7.219843E-04 | 5.838508E-06 | 7.494133E-07 |
| | 0.75 | 5.292500E-01 | 5.291809E-01 | 3.859379E-03 | 6.909595E-05 | 1.256582E-05 |
| | 1.00 | 6.795705E-01 | 6.791667E-01 | 1.290379E-02 | 4.037904E-04 | 9.224130E-05 |
| 0.75 | 0.25 | 7.222643E-01 | 7.222641E-01 | 9.632814E-05 | 1.977658E-07 | 1.340498E-08 |
| | 0.50 | 9.274057E-01 | 9.273926E-01 | 1.624465E-03 | 1.313664E-05 | 1.686180E-06 |
| | 0.75 | 1.190813E+0 | 1.190657E+0 | 8.683603E-03 | 1.554659E-04 | 2.827309E-05 |
| | 1.00 | 1.529034E+0 | 1.528125E+0 | 2.903353E-02 | 9.085285E-04 | 2.075429E-04 |

Table 7.4: Comparison with [206] for $\alpha = 1$ and extended FRDTM solutions of Example 7.5.2 with $\alpha = 0.8, 0.9$ taking first five terms

| $x$ | $t$ | Exact | HPM [206] | | extended FRDTM | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Approx. | $E_{abs}$ | Approx.($\alpha = 1$) | $E_{abs}(\alpha = 1)$ | $\alpha = 0.8$ | $\alpha = 0.9$ |
| 0.25 | 0.25 | 8.025159E-02 | 8.025106E-02 | 5.300000E-07 | 8.025163E-02 | 4.160431E-08 | 9.214191E-02 | 8.515460E-02 |
| | 0.50 | 1.030451E-01 | 1.030273E-01 | 1.773500E-05 | 1.030477E-01 | 2.609383E-06 | 1.278456E-01 | 1.129848E-01 |
| | 0.75 | 1.323125E-01 | 1.321716E-01 | 1.408700E-04 | 1.323416E-01 | 2.907458E-05 | 1.762209E-01 | 1.493639E-01 |
| | 1.00 | 1.698926E-01 | 1.692708E-01 | 6.217800E-04 | 1.700521E-01 | 1.594691E-04 | 2.428182E-01 | 1.974343E-01 |
| 0.50 | 0.25 | 3.210064E-01 | 3.210042E-01 | 2.123000E-06 | 3.210065E-01 | 1.664173E-07 | 3.685676E-01 | 3.406184E-01 |
| | 0.50 | 4.121803E-01 | 4.121094E-01 | 7.094300E-05 | 4.121906E-01 | 1.043753E-05 | 5.113825E-01 | 4.519393E-01 |
| | 0.75 | 5.292500E-01 | 5.286865E-01 | 5.634830E-04 | 5.293663E-01 | 1.162983E-04 | 7.048835E-01 | 5.974555E-01 |
| | 1.00 | 6.795705E-01 | 6.770833E-01 | 2.487123E-03 | 6.802083E-01 | 6.378762E-04 | 9.712730E-01 | 7.897371E-01 |
| 0.75 | 0.25 | 7.222643E-01 | 7.222595E-01 | 4.776000E-06 | 7.222646E-01 | 3.744388E-07 | 8.292771E-01 | 3.406184E-01 |
| | 0.50 | 9.274056E-01 | 9.272461E-01 | 1.596200E-04 | 9.274292E-01 | 2.348445E-05 | 1.150611E+0 | 4.519393E-01 |
| | 0.75 | 1.190813E+0 | 1.189545E+0 | 1.267830E-03 | 1.191075E+0 | 2.616713E-04 | 1.585988E+0 | 5.974555E-01 |
| | 1.00 | 1.529034E+0 | 1.523438E+0 | 5.596030E-03 | 1.530469E+0 | 1.435222E-03 | 2.185364E+0 | 7.897371E-01 |

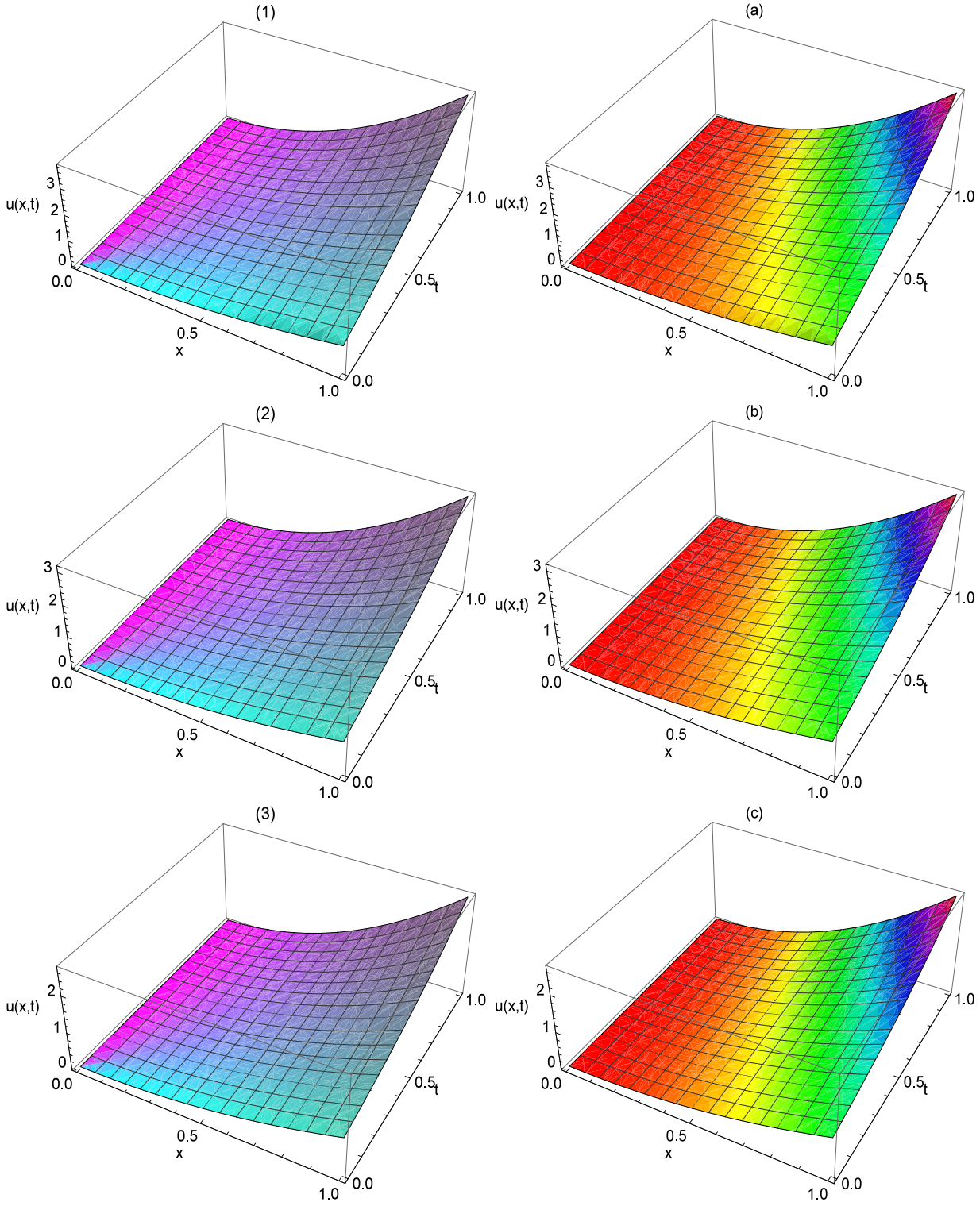


Figure 7.3: The solution behavior of AVIM solutions (a) to (c) and extended FRDTM solutions (1) to (3) of Example 7.5.2 for $\alpha = 0.8; 0.9; 1.0$ (top to bottom)

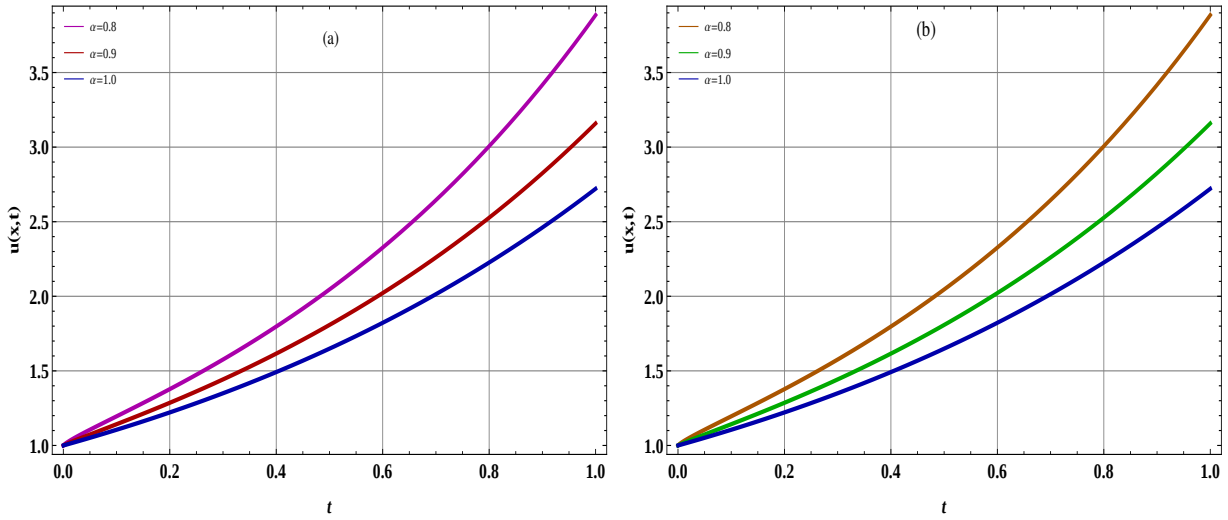


Figure 7.4: Two dimensional plots of $(a)$ AVIM solutions and $(b)$ extended FRDTM solutions of Example 7.5.2 for different values of $\alpha = 0.8, 0.9, 1.0$ at $t \in (0, 1); x = 1$

*The comparison of extended FRDTM solutions with HPM solutions [206] for* $\alpha = 1$ *and the extended FRDTM solutions for* $\alpha = 0.8, 0.9$ *are reported in Table 7.4. The behavior of AVIM solutions and the extended FRDTM solutions for different values of* $\alpha = 0.8, 0.9, 1.0$ *are depicted in Figure 7.3(a-c) and Fig.7.3(1-3). Figure 7.4 depicts the two dimensional plots of the AVIM solutions and extended FRDTM solutions for different values of* $\alpha = 0.8, 0.9, 1.0$ *at different time levels* $t \leq 1$ *and* $x = 1$*, the similar solution behaviors has been obtained by employing HPM [206].*

*In particular for* $\alpha = 1$*, both the AVIM solution* (7.5.12) *and extended FRDTM solution* (7.5.14) *reduces to*

$$u(x,t) = \left(1 + t + \frac{t^2}{2!} + \frac{t^3}{3!} + \ldots\right) x^2 \tag{7.5.15}$$

*The same results is obtained in [2] using DTM and RDTM, and converging to the exact result* $u(x,t) = x^2 \exp(t)$*. The findings shows that the computed AVIM solutions and extended FRDTM solutions of Example 7.5.2 are agreed well with solutions computed by employing HPM [206] and approaching to the exact solution.*

### 7.5.3 Solution of Example 7.5.3

**A. AVIM**

The iteration formula for (7.5.3) can be constructed as

$$\begin{cases} s_0 = x^2 \\ s_{k+1} = -\mathcal{J}_t^{\alpha}\left[\mathcal{D}_t^{\alpha}\sum_{i=0}^{k} s_i(x,t) + \sum_{i=0}^{k}\left\{s_i(x,t) + \frac{1}{8}\frac{\partial s_i(x,t)}{\partial x}\right\} + \sum_{i=0}^{k}\frac{\partial s_i\left(\frac{x}{2},\frac{t}{2}\right)}{\partial x}\frac{\partial^2 s_{k-i}\left(x,\frac{t}{2}\right)}{\partial x^2}\right] \end{cases} \tag{7.5.16}$$

Simplification of above relations leads to

$$\begin{cases} s_0 = x^2, \\ s_1 = -\dfrac{t^{\alpha}x^2}{\Gamma(\alpha+1)}, \\ s_2 = \dfrac{2^{-2+\alpha}t^{2\alpha}x(-2+2^{\alpha}(1+4x))}{\Gamma(1+2\alpha)}, \\ s_3 = \left\{\dfrac{8^{1+\alpha}x\Gamma(\frac{1}{2}\alpha) - \sqrt{\pi}\left((-2+2^{\alpha})(-2+4^{\alpha}) + 2^{4+\alpha}x(-1-2^{\alpha}+4^{\alpha}(1+2x))\Gamma(1+\alpha)\right)}{2^{5+3\alpha}\sqrt{\pi}\Gamma(1+\alpha)\Gamma(1+3\alpha)}\right\} \\ \qquad \times t^{3\alpha}, \\ \vdots \end{cases} \tag{7.5.17}$$

and so, the solution is given by

$$u(x,t) = s_0(x,t) + s_1(x,t) + s_2(x,t) + \dots \tag{7.5.18}$$

**B. Extended FRDTM**

Using extended FRDTM on (7.5.3), we get the following recurrence relation in $U_{\alpha}^{k}(x)$

$$\begin{cases} \dfrac{\Gamma\left(1+(1+k)\alpha\right)}{\Gamma\left(1+k\alpha\right)}U_{\alpha}^{k+1}(x) = \dfrac{1}{2^{k\alpha}}\displaystyle\sum_{r=0}^{k}\frac{\partial}{\partial x}U_{\alpha}^{k-r}\left(\frac{x}{2}\right)\frac{\partial^2}{\partial x^2}U_{\alpha}^{r}\left(\frac{x}{2}\right) - \frac{1}{8}\frac{\partial}{\partial x}U_{\alpha}^{k}(x) - U_{\alpha}^{k}(x) \\ U_{\alpha}^{0}(x) = x^2, \end{cases} \tag{7.5.19}$$

By solving the recurrence relation (7.5.19), we get

$$
\begin{aligned}
U_\alpha^0(x) &= x^2,\\
U_\alpha^1(x) &= -\frac{x^2}{\Gamma(1+\alpha)},\\
U_\alpha^2(x) &= \frac{x\left(1-2^{1-\alpha}+4x\right)}{4\Gamma(1+2\alpha)}\\
U_\alpha^3(x) &= \frac{-4+2^{1+\alpha}+2^{1+2\alpha}-8^\alpha-2^{4+\alpha}\left(-1-2^\alpha+4^\alpha\right)x-2^{5+3\alpha}x^2}{2^{5+3\alpha}\Gamma(1+3\alpha)}+\\
&\quad \frac{2^{-2-2\alpha}x\Gamma(1+2\alpha)}{\left(\Gamma(1+\alpha)\right)^2\Gamma(1+3\alpha)}\\
U_\alpha^4(x) &= \frac{\left(4+2^{2+\alpha}-2^{2+3\alpha}-4^{1+2\alpha}+3\ 32^\alpha+2^{3+2\alpha}\left(-2-2^{1+\alpha}-2^{1+2\alpha}+3\ 8^\alpha\right)x\right)}{32^{1+\alpha}\Gamma(1+4\alpha)}+\\
&\quad \frac{32^{1+\alpha}x^2}{32^{1+\alpha}\Gamma(1+4\alpha)}-\frac{\left(-2+8^\alpha+8^{1+\alpha}x\right)\left(\Gamma(1+2\alpha)\right)^2}{32^{1+\alpha}\left(\Gamma(1+\alpha)\right)^2\Gamma(1+2\alpha)\Gamma(1+4\alpha)}\\
&\quad -\frac{2^{1+\alpha}\left(-2+2^\alpha+2^{3+\alpha}x\right)\Gamma(1+\alpha)\Gamma(1+3\alpha)}{32^{1+\alpha}\left(\Gamma(1+\alpha)\right)^2\Gamma(1+2\alpha)\Gamma(1+4\alpha)}\\
&\quad \vdots \qquad\qquad \vdots
\end{aligned}
$$

On taking the inverse FRDTM, we get

$$
\begin{aligned}
u(x,t) &= x^2-\frac{x^2t^\alpha}{\Gamma(1+\alpha)}+\frac{x\left(1-2^{1-\alpha}+4x\right)}{4\Gamma(1+2\alpha)}t^{2\alpha}+\left\{\frac{-4+2^{1+\alpha}}{2^{5+3\alpha}\Gamma(1+3\alpha)}\right\}\\
&\quad +\left\{\frac{2^{1+2\alpha}-8^\alpha-2^{4+\alpha}\left(-1-2^\alpha+4^\alpha\right)x-2^{5+3\alpha}x^2}{2^{5+3\alpha}\Gamma(1+3\alpha)}+\frac{2^{-2-2\alpha}x\Gamma(1+2\alpha)}{\left(\Gamma(1+\alpha)\right)^2\Gamma(1+3\alpha)}\right\}t^{3\alpha}\\
&\quad +\left\{\frac{\left(4+2^{2+\alpha}-2^{2+3\alpha}-4^{1+2\alpha}+3\ 32^\alpha+2^{3+2\alpha}\left(-2-2^{1+\alpha}-2^{1+2\alpha}+3\ 8^\alpha\right)x+32^{1+\alpha}x^2\right)}{32^{1+\alpha}\Gamma(1+4\alpha)}\right.\\
&\quad \left.-\frac{\left(-2+8^\alpha+8^{1+\alpha}x\right)\left(\Gamma(1+2\alpha)\right)^2-2^{1+\alpha}\left(-2+2^\alpha+2^{3+\alpha}x\right)\Gamma(1+\alpha)\Gamma(1+3\alpha)}{32^{1+\alpha}\left(\Gamma(1+\alpha)\right)^2\Gamma(1+2\alpha)\Gamma(1+4\alpha)}\right\}t^{4\alpha}\\
&\quad +\ldots
\end{aligned}
\tag{7.5.20}
$$

**Remark 7.5.6.** *It is found from Table 7.5 and Table 7.6 that the computed results by both of the methods are the closed form of the results reported in [206]. The fifth order AVIM solutions for $\alpha = 1$ are compared with exact solutions and the absolute errors in different orders ($m = 3, 5, 7$) solutions are reported in Table 7.5. The comparison of extended*

Table 7.5: Approximate AVIM solution of Example 7.5.3 for $\alpha = 1$

| $x$ | $t$ | Exact | AVIM | | | |
|---|---|---|---|---|---|---|
| | | | Approx. | $\lvert u - u_3 \rvert$ | $\lvert u - u_5 \rvert$ | $\lvert u - u_7 \rvert$ |
| 0.25 | 0.25 | 4.867505E-02 | 4.867503E-02 | 9.684359E-06 | 2.045889E-08 | 3.031331E-09 |
| | 0.50 | 3.790817E-02 | 3.790690E-0 | 1.477496E-04 | 1.265190E-06 | 1.364865E-07 |
| | 0.75 | 2.952291E-02 | 2.950897E-02 | 7.143158E-04 | 1.393738E-05 | 9.287844E-07 |
| | 1.00 | 2.299247E-02 | 2.291667E-02 | 2.159132E-03 | 7.579841E-05 | 1.853830E-06 |
| 0.50 | 0.25 | 1.947002E-01 | 1.947001E-01 | 3.873743E-05 | 8.183556E-08 | 6.175209E-09 |
| | 0.50 | 1.516327E-01 | 1.516276E-01 | 5.909983E-04 | 5.060761E-06 | 2.929576E-07 |
| | 0.75 | 1.180916E-01 | 1.180359E-01 | 2.857263E-03 | 5.574951E-05 | 2.289376E-06 |
| | 1.00 | 9.196986E-02 | 9.166667E-02 | 8.636527E-03 | 3.031936E-04 | 7.584863E-06 |
| 0.75 | 0.25 | 4.380754E-01 | 4.380753E-01 | 8.715923E-05 | 1.841300E-07 | 9.365110E-09 |
| | 0.50 | 3.411735E-01 | 3.411621E-01 | 1.329746E-03 | 1.138671E-05 | 4.608981E-07 |
| | 0.75 | 2.657062E-01 | 2.655807E-01 | 6.428842E-03 | 1.254364E-04 | 3.936288E-06 |
| | 1.00 | 2.069322E-01 | 2.062500E-01 | 1.943219E-02 | 6.821857E-04 | 1.610319E-05 |

Table 7.6: Comparison with HPM [206] for $\alpha = 1$ and extended FRDTM solutions of Example 7.5.3 with $\alpha = 0.8, 0.9$ taking first five terms

| $x$ | $t$ | Exact | HPM [206] | | Extended FRDTM | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Approx. | $E_{abs}$ | Approx.($\alpha = 1$) | $E_{abs}(\alpha = 1)$ | $\alpha = 0.8$ | $\alpha = 0.9$ |
| 0.25 | 0.25 | 4.8675049E-02 | 4.867554E-02 | 4.880000E-07 | 4.867503E-02 | 2.045889E-08 | 4.3977917E-02 | 4.6422026E-02 |
| | 0.50 | 3.7908166E-02 | 3.792318E-02 | 1.501090E-05 | 3.790690E-02 | 1.265190E-06 | 3.4060152E-02 | 3.5961890E-02 |
| | 0.75 | 2.9522910E-02 | 2.963257E-02 | 1.096588E-04 | 2.950897E-02 | 1.393738E-05 | 2.7211675E-02 | 2.8278279E-02 |
| | 1.00 | 2.2992465E-02 | 2.343750E-02 | 4.450349E-04 | 2.291667E-02 | 7.579841E-05 | 2.2236160E-02 | 2.2416961E-02 |
| 0.50 | 0.25 | 1.9470020E-01 | 1.947022E-01 | 1.952000E-06 | 1.947001E-01 | 8.183556E-08 | 1.7684986E-01 | 1.8601398E-01 |
| | 0.50 | 1.5163266E-01 | 1.516927E-01 | 6.004300E-05 | 1.516276E-01 | 5.060762E-06 | 1.3830562E-01 | 1.4471678E-01 |
| | 0.75 | 1.1809164E-01 | 1.185303E-01 | 4.386360E-04 | 1.180359E-01 | 5.574951E-05 | 1.1159170E-01 | 1.1447771E-01 |
| | 1.00 | 9.1969860E-02 | 9.375000E-02 | 1.780140E-03 | 9.166666E-02 | 3.031936E-04 | 9.1384701E-02 | 9.1285983E-02 |
| 0.75 | 0.25 | 4.3807544E-01 | 4.380798E-01 | 4.394000E-06 | 4.380752E-01 | 1.841300E-07 | 3.9862936E-01 | 4.1878006E-01 |
| | 0.50 | 3.4117350E-01 | 3.413086E-01 | 1.350980E-04 | 3.411621E-01 | 1.138671E-05 | 3.1279528E-01 | 3.2628721E-01 |
| | 0.75 | 2.6570619E-01 | 2.666931E-01 | 9.869290E-04 | 2.655806E-01 | 1.254364E-04 | 2.5328325E-01 | 2.5865725E-01 |
| | 1.00 | 2.0693219E-01 | 2.109375E-01 | 4.005314E-03 | 2.062500E-01 | 6.821857E-04 | 2.0773733E-01 | 2.0672821E-01 |

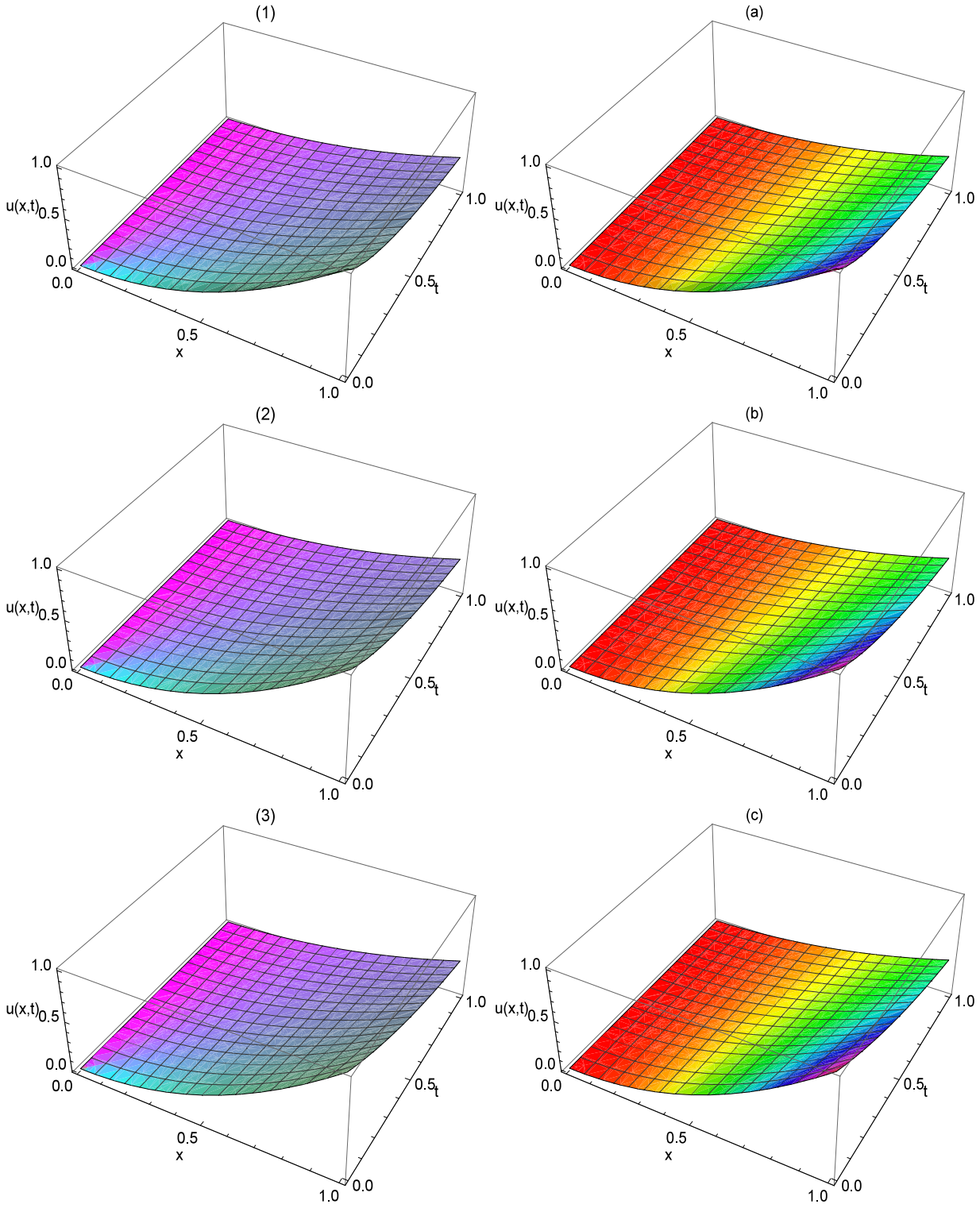


Figure 7.5: The solution behavior of AVIM solutions (a) to (c) and extended FRDTM solutions (1) to (3) of Example 7.5.3 for $\alpha = 0.8; 0.9; 1.0$ (top to bottom)

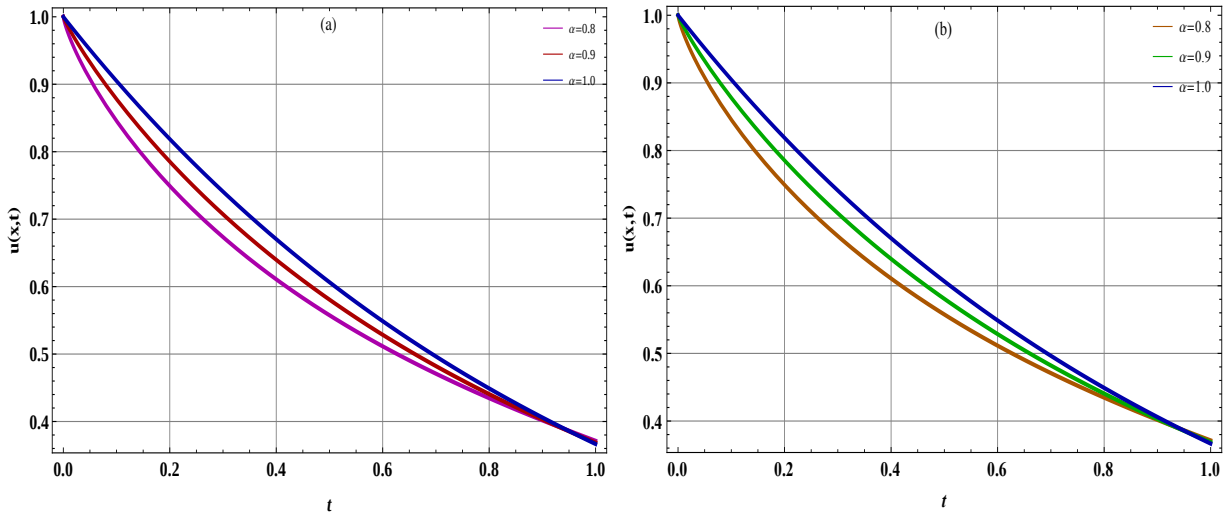


Figure 7.6: Two dimensional plots of $(a)$ AVIM solutions and $(b)$ extended FRDTM solutions of Example 7.5.3 for different values of $\alpha = 0.8, 0.9, 1.0$ at $t \in (0, 1); x = 1$

*FRDTM solutions with HPM solutions [206] for* $\alpha = 1$ *and the extended FRDTM solutions for* $\alpha = 0.8, 0.9$ *are reported in Table 7.6. The behavior of AVIM solutions and the extended FRDTM solutions for different values of* $\alpha = 0.8, 0.9, 1.0$ *are depicted in Figure 7.5.Figure 7.6 depicts the two dimensional plots of the AVIM solutions and extended FRDTM solutions for different values of* $\alpha = 0.8, 0.9, 1.0$ *at different time levels* $t \leq 1$ *and* $x = 1$*, the similar solution behaviors has been obtained by employing HPM [206].*

*In particular for* $\alpha = 1$*, both the AVIM solution* (7.5.18) *and extended FRDTM solution* (7.5.20) *reduces to*

$$u(x,t) = \left(1 - t + \frac{t^2}{2} - \frac{t^3}{6} + \frac{t^4}{24} - \dots\right) x^2. \tag{7.5.21}$$

*The same result is obtained in [2] using DTM and RDTM, and converging to the exact results* $u(x,t) = x^2 \exp(-t)$*. The findings shows that the computed AVIM solutions and extended FRDTM solutions of Example 7.5.3 are agreed well with solutions computed by employing HPM [206] and approaching to the exact solution.*

## 7.6 Conclusion

In this chapter, some basic properties of fractional reduced differential transform method extended for time-fractional partial differential equations with proportional delay. Furthermore, the approximate solutions of time-fractional partial differential equations with proportional delay are obtained by employing two reliable methods: `alternative variation iteration method` and `extended fractional reduced differential transform method`. The efficiency and validity of these methods are illustrated by considering three examples. The findings are as follows:

- ✓ The approximate analytical results have been given in terms of a power series.
- ✓ The proposed solutions by both of the methods converges to the exact solutions. The computation of extended FRDTM solutions are easy in comparison to AVIM.
- ✓ The computed AVIM solutions and extended FRDTM solutions are agreed excellently with HPM [206], HPTM [245] and DTM [2].
- ✓ These approximate solutions are obtained without any discretization or perturbation.

# Chapter 8

# Homotopy perturbation transform method for solving fractional PDEs with proportional delay

## 8.1 Introduction

**Definition 8.1.1** ( [120, 166]). *The Laplace transform of a piecewise continuous function* $u(t)$ *in* $(0, \infty)$ *is defined by*

$$\mathcal{U}(s) = \mathcal{L}\{u(t)\} = \int_0^\infty u(t) \exp(-st) dt, \tag{8.1.1}$$

*where* $s$ *is a parameter.*

*Moreover, for the Caputo derivative* $\mathcal{D}_t^\alpha u\,(t)$ *and Riemann– Liouville fractional integral* $\mathcal{J}_t^\alpha u\,(t)$ *of a function* $u \in \mathbb{C}_\mu$ $(\mu \geq -1)$*, the Laplace transform is defined as*

$$\begin{aligned} &\mathcal{L}\{\mathcal{J}_t^\alpha u\,(t)\} = s^{-\alpha}\mathcal{U}(s), \\ &\mathcal{L}\{\mathcal{D}_t^\alpha f\,(t)\} = s^\alpha \mathcal{U}(s) - \sum_{r=0}^{m-1} s^{\alpha-r-1} u^{(r)}(0+), \quad (m-1 < \alpha \leq m) \end{aligned} \tag{8.1.2}$$

In the past years, vigourous techniques with the Laplace transform have been developed [74, 112, 115–119, 136], among them, HPTM has been employed for solving fractional

model of Navier-Stokes equation [139], optimal control problems [69], fractional coupled sine-Gordon equations [203], Falkner-Skan wedge flow [161], time- and space-fractional coupled Burgers' equations [243], strongly nonlinear oscillators [179], non-homogeneous partial differential equations with a variable coefficient [160] and many more.

The main goal of this chapter is the new application homotopy perturbation transform method (HPTM) in the numerical study of the initial value autonomous system of time fractional partial differential equations (TFPDEs) with proportional delay (8.2.1).

## 8.2 HPTM for TFPDEs with proportional delay

This section deals with the implementation of HPTM to the following initial value autonomous system of time fractional partial differential equations with proportional delay as in (7.4.5)

$$\begin{cases} \mathcal{D}_t^{\alpha}\left(u(x,t)\right) = f\left(x, u(a_0x,b_0t), \frac{\partial}{\partial x}u(a_1x,b_1t), \ldots, \frac{\partial^m}{\partial x^m}u(a_mx,b_mt)\right), \\ u(x,0) = \psi(x). \end{cases} \tag{8.2.1}$$

Laplace transform of equation (8.2.1) yields the following

$$\mathcal{U}(x,s) = \frac{u(x,0)}{s} + \frac{1}{s^{\alpha}}\mathcal{L}\left\{f\left(x, u(a_0x,b_0t), \frac{\partial}{\partial x}u(a_1x,b_1t), \ldots, \frac{\partial^m}{\partial x^m}u(a_mx,b_mt)\right)\right\} \tag{8.2.2}$$

The inverse Laplace transform of equation (8.2.2) yields the following

$$\begin{aligned} u(x,t) &= \psi(x) \\ &\quad + \mathcal{L}^{-1}\left\{\frac{1}{s^{\alpha}}\mathcal{L}\left\{f\left(x, u(a_0x,b_0t), \frac{\partial}{\partial x}u(a_1x,b_1t), \ldots, \frac{\partial^m}{\partial x^m}u(a_mx,b_mt)\right)\right\}\right\}, \end{aligned} \tag{8.2.3}$$

where $\psi(x)$ is due to the initial conditions.

The homotopy for equation (8.2.3) is defined as follows

$$\begin{aligned} u(x,t) &= \psi(x) \\ &\quad + p\mathcal{L}^{-1}\left\{\frac{1}{s^{\alpha}}\mathcal{L}\left\{f\left(x, u(a_0x,b_0t), \frac{\partial}{\partial x}u(a_1x,b_1t), \ldots, \frac{\partial^m}{\partial x^m}u(a_mx,b_mt)\right)\right\}\right\}, \end{aligned} \tag{8.2.4}$$

Let the basic solution of Equation (8.2.1), i.e., Equation (8.2.4) read as

$$u(x,t)=\sum_{\imath=0}^{\infty}p^{\imath}u_{\imath}(x,t). \tag{8.2.5}$$

and so, nonlinear term $N[u(x,t)]$ (let) occurred in the right hand side of equation (8.2.1) is decomposed as follows

$$N[u(x,t)]=\sum_{\ell=0}^{\infty}p^{\ell}H_{\ell}(u),\quad H_{\ell}(u)=\frac{1}{\ell!}\left\{\frac{\partial^{\ell}}{\partial p^{\ell}}N\left[\sum_{i=0}^{\infty}p^{i}u_{i}(x,t)\right]\right\}_{p=0} \tag{8.2.6}$$

where $H_{\ell}(u)\equiv H_{\ell}\left(u_0(x,t),u_1(x,t),\ldots,u_{\ell}(x,t)\right)$ is referred to as He's polynomial [86].

From Equation (8.2.5) and (8.2.3), we get

$$\begin{aligned}\sum_{r=0}^{\infty}p^{r}u_r(x,t)=u(x,0)+p\Bigg[\mathcal{L}^{-1}\Bigg\{\frac{1}{s^{\alpha}}\mathcal{L}\Bigg\{f\Bigg(x,t,\sum_{r=0}^{\infty}p^{r}u_r(a_0x,b_0t),\\ \frac{\partial}{\partial x}\sum_{r=0}^{\infty}p^{r}u_r(a_1x,b_1t),\ldots\frac{\partial^{m}}{\partial x^{m}}\sum_{r=0}^{\infty}p^{r}u_r(a_mx,b_mt)\Bigg)\Bigg\}\Bigg\}\Bigg]\end{aligned} \tag{8.2.7}$$

On equating like powers of $p$, we get

$$\begin{aligned}&p^0:u_0(x,t)=\psi(x)\\ &p^1:u_1(x,t)=\mathcal{L}^{-1}\left[\frac{1}{s^{\alpha}}\mathcal{L}\left[\underline{f}\left(x,t,u_0(a_0x,b_0t),\frac{\partial}{\partial x}u_0(a_1x,b_1t),\ldots,\frac{\partial^{m}}{\partial x^{m}}u_0(a_mx,b_mt)\right)\right]\right]\\ &\qquad\vdots\qquad\qquad\qquad\vdots\end{aligned}$$

Consequently, the $m$th order approximate solution of equation (8.2.1) is corresponding to $p=1$ as follows

$$s_m=\sum_{\imath=0}^{m}u_{\imath}(x,t). \tag{8.2.8}$$

and so, $s_m$ converges to the exact solution of equation (8.2.1). That is,

$$u(x,t)=\lim_{m\to\infty}s_m=\sum_{\imath=0}^{\infty}u_{\imath}(x,t). \tag{8.2.9}$$

**Theorem 8.2.1.** *[206, Theorem 4.1, 4.2] Let $0<\gamma<1$ and let $u_n(x,t),u(x,t)$ are in*

*Banach space* $(\mathcal{C}[0,1], ||\cdot||)$. *Then*

*a) The series solution* $\sum_{n=0}^{\infty} u_n(x,t)$ *from the sequence* $\{u_n(x)\}_{n=0}^{\infty}$ *converges to the solution of equation* (8.2.1) *whenever* $u_n(x) \leq \gamma u_{n-1)}(x)$ *for all* $n \in N$.

*b) The maximum absolute truncation error of the series solution* (8.2.9) *for equation* (8.2.1) *is computed as*

$$\left|\left|u(x,t) - \sum_{\imath=0}^{\ell} u_\imath(x,t)\right|\right| \leq \frac{\gamma^{\ell+1}}{1-\gamma}||u_0(x,t)||. \tag{8.2.10}$$

## 8.3 Application of HPTM for TFPDEs with proportional delay

This section deals with the effectiveness and validity of HPTM, illustrated by three test problems of initial value autonomous system of TFPDEs with proportional delay.

**Example 8.3.1.** *The first case deals with initial value system of time-fractional order generalized Burgers equation with proportional delay [206] as*

$$\begin{cases} \mathcal{D}_t^{\alpha} u(x,t) = \dfrac{\partial^2}{\partial x^2} u(x,t) + u\left(\dfrac{x}{2}, \dfrac{t}{2}\right) \dfrac{\partial}{\partial x} u\left(x, \dfrac{t}{2}\right) + \dfrac{1}{2} u(x,t) \\ u(x,0) = x, \end{cases} \tag{8.3.1}$$

*Taking Laplace transformation of Eq.* (8.3.1), *we get*

$$\mathcal{U}(x,s) = \frac{x}{s} + \frac{1}{s^{\alpha}} \mathcal{L}\left[\frac{\partial^2}{\partial x^2} u(x,t) + u\left(\frac{x}{2}, \frac{t}{2}\right) \frac{\partial}{\partial x} u\left(x, \frac{t}{2}\right) + \frac{1}{2} u(x,t)\right] \tag{8.3.2}$$

*Now, inverse Laplace transform with basic solution* (8.2.5) *leads to*

$$\begin{aligned} \sum_{r=0}^{\infty} p^r u_r(x,t) = x + p \sum_{r=0}^{\infty} p^r \Bigg[\mathcal{L}^{-1}\Bigg\{\frac{1}{s^{\alpha}} \mathcal{L}\Bigg\{\frac{\partial^2 u_r(x,t)}{\partial x^2} \\ + \frac{1}{2} u_r(x,t) + \sum_{k=0}^{r} u_k\left(\frac{x}{2}, \frac{t}{2}\right) \frac{\partial u_{r-k}\left(x, \frac{t}{2}\right)}{\partial x}\Bigg\}\Bigg\}\Bigg] \end{aligned} \tag{8.3.3}$$

*On comparing the coefficient of like powers of* $p^r$ *on both sides of Equation* (8.3.3)*, we get*

$$\begin{cases} \text{Coefficient of } p^0: \ u_0(x,t) = x, \\ \text{Coefficient of } p^{r+1}: u_{r+1}(x,t) = \mathcal{L}^{-1}\left\{\dfrac{1}{s^\alpha}\mathcal{L}\left\{\dfrac{\partial^2 u_r(x,t)}{\partial x^2}\right.\right. \\ \qquad\qquad \left.\left. +\dfrac{1}{2}u_r(x,t) + \displaystyle\sum_{k=0}^{r} u_k\left(\frac{x}{2},\frac{t}{2}\right)\frac{\partial u_{r-k}\left(x,\frac{t}{2}\right)}{\partial x}\right\}\right\}, r \geq 0. \end{cases} \tag{8.3.4}$$

*Recurrence relation* (8.3.4) *yields*

$$\begin{aligned} u_0(x,t) &= x, \\ u_1(x,t) &= \frac{xt^\alpha}{\Gamma(1+\alpha)} \\ u_2(x,t) &= \frac{(1+2^{1-\alpha})xt^{2\alpha}}{2\Gamma(1+2\alpha)} \\ u_3(x,t) &= \frac{xt^{3\alpha}}{4\Gamma(1+3\alpha)}\left\{1+2^{1-\alpha}+2^{1-2\alpha}+2^{2-3\alpha}+\frac{\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}\,2^{1-2\alpha}\right\} \\ u_4(x,t) &= \frac{xt^{4\alpha}}{8\Gamma(1+4\alpha)}\left\{1+2^{9-6\alpha}+2^{8-5\alpha}+3\times 2^{7-3\alpha}+2^{7-2\alpha}+2^{7-\alpha}+2^{8-4\alpha}\right. \\ &\quad \left.+(2^{8-5\alpha}+2^{7-2\alpha})\frac{\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}+(2^{9-4\alpha}+2^{8-3\alpha})\frac{\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)}\right\},\ldots \\ &\quad\ \vdots \qquad\qquad\qquad \vdots \end{aligned} \tag{8.3.5}$$

*Therefore, the* $m$*th order approximate solution of equation* (8.3.1) *is*

$$u(x,t) = u_0(x,t) + u_1(x,t) + u_2(x,t) + u_3(x,t) + u_4(x,t) + \ldots + u_m(x,t) \tag{8.3.6}$$

*The same solution is obtained by Sarkar et al. [206]. In particular for* $\alpha = 1$*, the seventh order solution is*

$$u(x,t) = x\left(1 + t + \frac{t^2}{2} + \frac{t^3}{6} + \frac{t^4}{24} + \frac{t^5}{120} + \frac{t^6}{720} + \frac{t^7}{5040}\right) \tag{8.3.7}$$

*which is same as obtained by DTM and RDTM [2], and is a closed form of the exact solution* $u(x,t) = x\exp(t)$*. The comparison of the fifth order HPTM solutions with exact solutions and absolute errors in different order HPTM solutions* ($m = 4,5,6,7$) *for* $\alpha = 1$

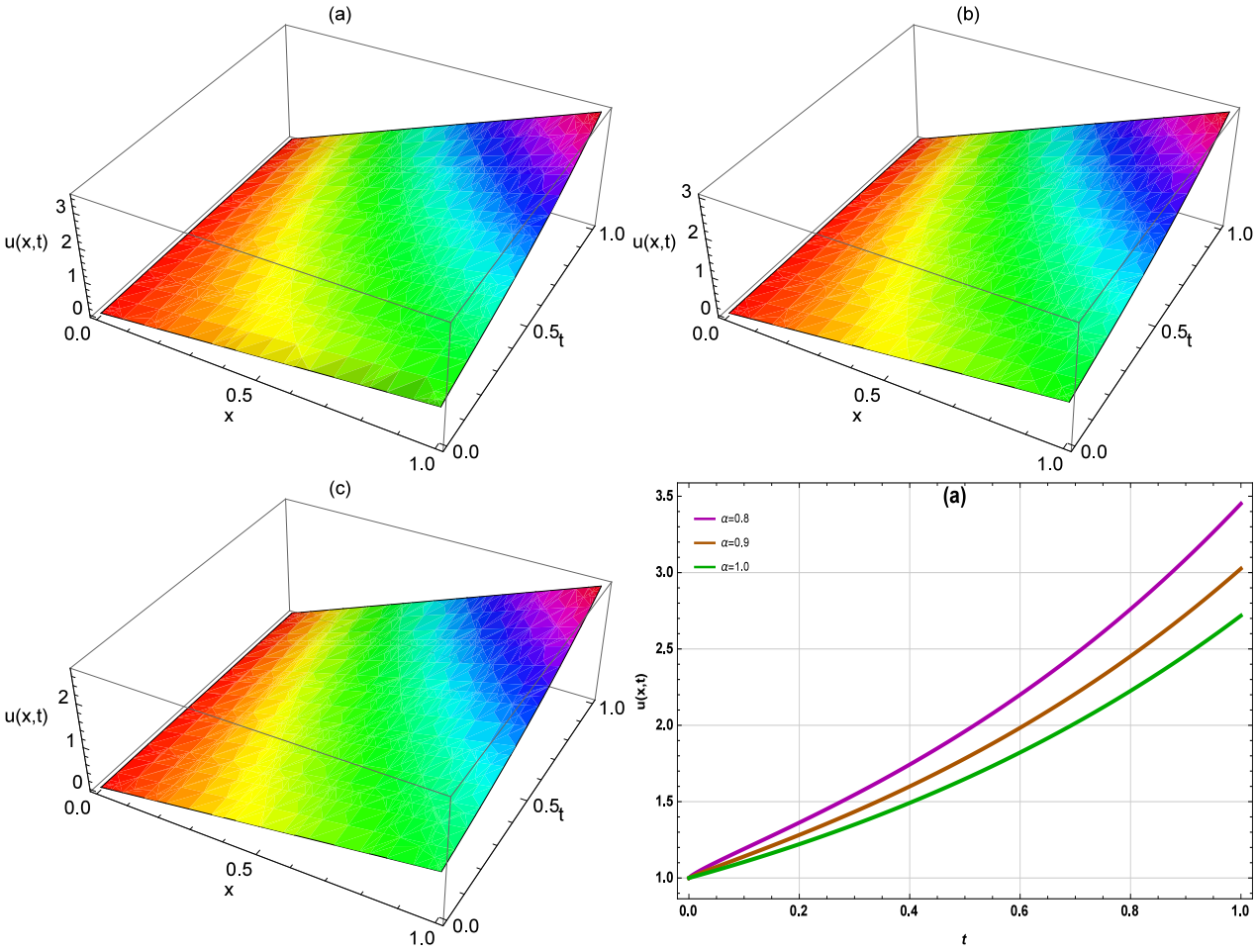


Figure 8.1: The surface HPTM solution behavior of $u$ of Example 8.3.1 for (a) $\alpha = 0.8$; (b ) $\alpha = 0.9$; (c) $\alpha = 1.0$; and their plots for different values of $\alpha = 0.8, 0.9, 1.0$ at different time levels $t \in (0, 1)$ and $x = 1$

*are reported in Table 8.1. The surface solution behavior of $u(x,t)$ for different values of $\alpha = 0.8, 0.9, 1.0$, and the plots of the solution for $x = 1$ at different time intervals $t \leq 1$ are depicted in Figure 8.1. The findings from Table 8.1 that the results are agreed well with the results obtained by HPM and DTM and approaching to the exact solution.*

**Example 8.3.2.** *Consider initial value TFPDE with proportional delay as given in [2, 206]*

$$\begin{cases} \mathcal{D}_t^{\alpha} u(x,t) = u\left(x, \dfrac{t}{2}\right) \dfrac{\partial^2}{\partial x^2} u\left(x, \dfrac{t}{2}\right) - u(x,t) \\ u(x,0) = x^2, \end{cases} \tag{8.3.8}$$

Table 8.1: Absolute errors in different order HPTM solutions ($m = 4, 5, 6, 7$) of Example 8.3.1 at $\alpha = 1.0$

| $x$ | $t$ | Exact | HPTM Solution | absolute errors in $m$th order HPTM solutions $\lvert u - s_4\rvert$ | $\lvert u - s_5\rvert$ | $\lvert u - s_6\rvert$ | $\lvert u - s_7\rvert$ |
|---|---|---|---|---|---|---|---|
| 0.25 | 0.25 | 0.321006 | 0.321004 | 2.1224E-06 | 8.7896E-08 | 3.1248E-09 | 9.7306E-11 |
| | 0.50 | 0.412180 | 0.412109 | 7.0943E-05 | 5.8385E-06 | 4.1316E-07 | 2.5636E-08 |
| | 0.75 | 0.529250 | 0.528686 | 5.6348E-04 | 6.9096E-05 | 7.2979E-06 | 6.7663E-07 |
| | 1.00 | 0.679570 | 0.677083 | 2.4871E-03 | 4.0379E-04 | 5.6568E-05 | 6.9651E-06 |
| 0.50 | 0.25 | 0.642012 | 0.642008 | 4.2448E-06 | 1.7579E-07 | 6.2497E-09 | 1.9461E-10 |
| | 0.50 | 0.824361 | 0.824219 | 1.4189E-04 | 1.1677E-05 | 8.2632E-07 | 5.1273E-08 |
| | 0.75 | 1.058500 | 1.057373 | 1.1270E-03 | 1.3819E-04 | 1.4596E-05 | 1.3533E-06 |
| | 1.00 | 1.359141 | 1.354167 | 4.9742E-03 | 8.0758E-04 | 1.1314E-04 | 1.3930E-05 |
| 0.75 | 0.25 | 0.963019 | 0.963012 | 6.3697E-06 | 2.6369E-07 | 9.3745E-09 | 2.9192E-10 |
| | 0.50 | 1.236541 | 1.236328 | 2.1283E-04 | 1.7516E-05 | 1.2395E-06 | 7.6909E-08 |
| | 0.75 | 1.587750 | 1.586060 | 1.6900E-03 | 2.0729E-04 | 2.1894E-05 | 2.0299E-06 |
| | 1.00 | 2.038711 | 2.031250 | 7.4614E-03 | 1.2114E-03 | 1.6970E-04 | 2.0895E-05 |

*Taking Laplace transform of Eq.* (8.3.8)*, we get*

$$\mathcal{U}(x,s) = \frac{x^2}{s} + \frac{1}{s^\alpha}\mathcal{L}\left[u\left(x,\frac{t}{2}\right)\frac{\partial^2}{\partial x^2}u\left(x,\frac{t}{2}\right) - u(x,t)\right] \tag{8.3.9}$$

*The inverse Laplace transform of Eq.* (8.3.9) *leads to*

$$u(x,t) = x^2 + \mathcal{L}^{-1}\left[\frac{1}{s}x + \frac{1}{s^\alpha}\mathcal{L}\left\{u\left(x,\frac{t}{2}\right)\frac{\partial^2}{\partial x^2}u\left(x,\frac{t}{2}\right) - u(x,t)\right\}\right] \tag{8.3.10}$$

*Eq.* (8.3.10) *with basic solution* (8.2.5) *leads to*

$$\begin{aligned}\sum_{n=0}^{\infty} p^n u_n(x,t) &= x^2 \\ &+ p\sum_{n=0}^{\infty} p^n\left[\mathcal{L}^{-1}\left\{\frac{1}{s^\alpha}\mathcal{L}\left(\sum_{k=0}^{n} u_k\left(x,\frac{t}{2}\right)\frac{\partial^2}{\partial x^2}u_{n-k}\left(x,\frac{t}{2}\right) - u_n(x,t)\right)\right\}\right]\end{aligned} \tag{8.3.11}$$

*On comparing the coefficient of like powers of $p^n$ on both sides of Equation* (8.3.11)*, we*

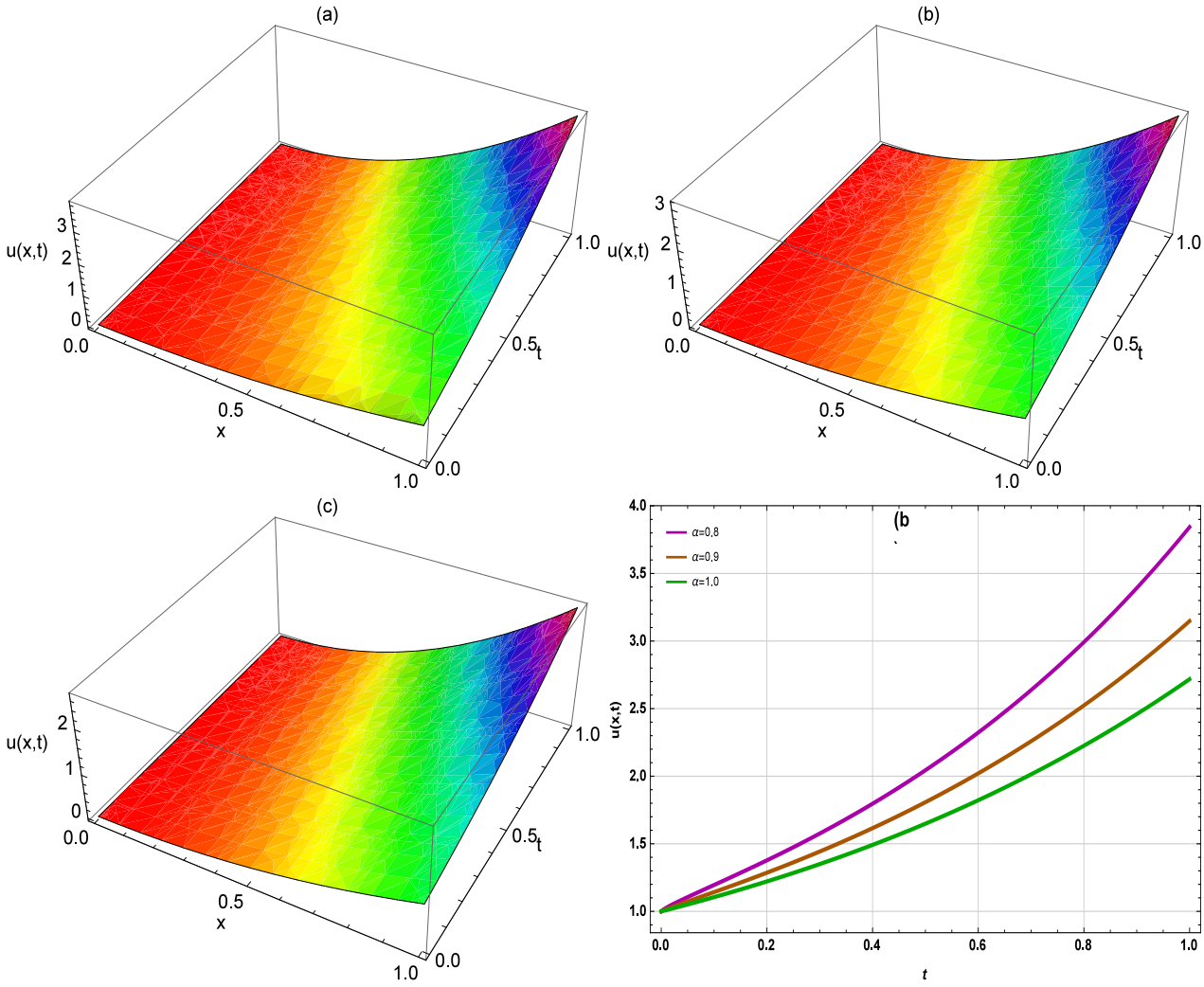


Figure 8.2: The surface HPTM solution behavior of $u$ of Example 8.3.2 for (a) $\alpha = 0.8$, (b ) $\alpha = 0.9$, (c) $\alpha = 1.0$; and their plots for different values of $\alpha = 0.8, 0.9, 1.0$ at different time levels $t \in (0, 1)$ and $x = 1$

*get*

$$\begin{cases} \textit{Coefficient of } p^0: \ u_0(x,t) = x^2, \\ \textit{Coefficient of } p^{n+1}: \\ u_{n+1}(x,t) = \mathcal{L}^{-1}\left\{\dfrac{1}{s^\alpha}\mathcal{L}\left(\displaystyle\sum_{k=0}^{n} u_k\left(x,\frac{t}{2}\right)\frac{\partial^2}{\partial x^2}u_{n-k}\left(x,\frac{t}{2}\right) - u_n(x,t)\right)\right\}, n \geq 0. \end{cases} \tag{8.3.12}$$

Table 8.2: Absolute errors in different order HPTM solutions ($m = 4, 5, 6, 7$) of Example 8.3.2 at $\alpha = 1.0$

| $x$ | $t$ | Exact | HPTM Solution | absolute errors in $m$th order HPTM solutions $\lvert u - s_4\rvert$ | $\lvert u - s_5\rvert$ | $\lvert u - s_6\rvert$ | $\lvert u - s_7\rvert$ |
|---|---|---|---|---|---|---|---|
| 0.25 | 0.25 | 0.080252 | 0.080252 | 5.3060E-07 | 2.1974E-08 | 7.8121E-10 | 2.4326E-11 |
| | 0.50 | 0.103045 | 0.103045 | 1.7736E-05 | 1.4596E-06 | 1.0329E-07 | 6.4091E-09 |
| | 0.75 | 0.132313 | 0.132311 | 1.4087E-04 | 1.7274E-05 | 1.8245E-06 | 1.6916E-07 |
| | 1.00 | 0.169893 | 0.169879 | 6.2178E-04 | 1.0095E-04 | 1.4142E-05 | 1.7413E-06 |
| 0.50 | 0.25 | 0.321006 | 0.321006 | 2.1224E-06 | 8.7896E-08 | 3.1248E-09 | 9.7306E-11 |
| | 0.50 | 0.412180 | 0.412180 | 7.0943E-05 | 5.8385E-06 | 4.1316E-07 | 2.5636E-08 |
| | 0.75 | 0.529250 | 0.529243 | 5.6348E-04 | 6.9096E-05 | 7.2979E-06 | 6.7663E-07 |
| | 1.00 | 0.679571 | 0.679514 | 2.4871E-03 | 4.0379E-04 | 5.6568E-05 | 6.9651E-06 |
| 0.75 | 0.25 | 0.722264 | 0.722264 | 4.7754E-06 | 1.9777E-07 | 7.0309E-09 | 2.1894E-10 |
| | 0.50 | 0.927406 | 0.927405 | 1.5962E-04 | 1.3137E-05 | 9.2961E-07 | 5.7682E-08 |
| | 0.75 | 1.190813 | 1.190796 | 1.2678E-03 | 1.5547E-04 | 1.6420E-05 | 1.5224E-06 |
| | 1.00 | 1.529034 | 1.528906 | 5.5960E-03 | 9.0853E-04 | 1.2728E-04 | 1.5671E-05 |

$$
\begin{aligned}
p^0 &: u_0(x,t) = x^2,\\
p^1 &: u_1(x,t) = \frac{x^2 t^{\alpha}}{\Gamma(1+\alpha)}\\
p^2 &: u_2(x,t) = \frac{x^2 t^{2\alpha}(2^{2-\alpha}-1)}{\Gamma(1+2\alpha)}\\
p^3 &: u_3(x,t) = \frac{x^2 t^{3\alpha}}{\Gamma(1+3\alpha)}\left\{1 - 2^{2-\alpha} - 2^{2-2\alpha} + 2^{4-3\alpha} + \frac{\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}\ 2^{1+\alpha}\right\}\\
p^4 &: u_4(x,t) = \frac{x^2 t^{4\alpha}}{\Gamma(1+4\alpha)}\left(1 - 2^{2-\alpha} - 2^{2-2\alpha} + 3\ 2^{2-3\alpha} + 2^{4-4\alpha} + 2^{4-5\alpha} - 2^{6-6\alpha}\right)\\
&\quad + \frac{x^2 t^{4\alpha}}{\Gamma(1+4\alpha)}\left(\frac{(2^{1+4\alpha} - 2^{3+2\alpha})\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2} + \frac{(2^{2+3\alpha} - 2^{4+2\alpha})\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)}\right)\\
&\qquad\qquad \vdots \qquad\qquad\qquad \vdots
\end{aligned}
$$

*Thus, the $m$th order approximate solution for Equation* (8.3.8) *is given by*

$$u(x,t) = u_0(x,t) + u_1(x,t) + u_2(x,t) + u_3(x,t) + u_4(x,t) + \ldots + u_m(x,t). \tag{8.3.13}$$

*Sarkar et al [206] obtained the same solution by employing HPM. In particular for $\alpha = 1$,*

*the seventh order solution is given by*

$$u(x,t) = x^2\left(1 + t + \frac{t^2}{2} + \frac{t^3}{6} + \frac{t^4}{24} + \frac{t^5}{120} + \frac{t^6}{720} + \frac{t^7}{5040}\right), \tag{8.3.14}$$

*which is same as obtained by DTM and RDTM [2], and is a closed form of the exact solution* $u(x,t) = x^2\exp(t)$. *The HPTM solutions for* $\alpha = 1$ *are compared with the exact solutions and the absolute errors in different order HPTM solutions are reported in Table 8.2. The findings show that the proposed HPTM results are agreed well with HPM and DTM solutions and approaching to the exact solutions. The surface solution behavior of* $u(x,t)$ *for different values of* $\alpha = 0.8, 0.9, 1.0$, *and the plots of the solution for* $x = 1$ *at different time intervals* $t \leq 1$ *are depicted in Figure 8.2.*

**Example 8.3.3.** *Consider initial value TFPDE with proportional delay as given in [2, 206]*

$$\begin{cases} \mathcal{D}_t^{\alpha} u(x,t) = \dfrac{\partial^2}{\partial x^2} u\left(\dfrac{x}{2}, \dfrac{t}{2}\right) \ \dfrac{\partial}{\partial x} u\left(\dfrac{x}{2}, \dfrac{t}{2}\right) - \dfrac{1}{8}\dfrac{\partial}{\partial x} u\,(x,t) - u(x,t) \\ u(x,0) = x^2, \end{cases} \tag{8.3.15}$$

*The Laplace transform of Equation* (8.3.15) *yields the following*

Table 8.3: Absolute errors in different order HPTM solutions ($m = 4, 5, 6, 7$) of Example 8.3.3 at $\alpha = 1.0$

| $x$ | $t$ | Exact | HPTM Solution | absolute errors in $m$th order HPTM solutions $\lvert u - s_4\rvert$ | $\lvert u - s_5\rvert$ | $\lvert u - s_6\rvert$ | $\lvert u - s_7\rvert$ |
|---|---|---|---|---|---|---|---|
| 0.25 | 0.25 | 0.0486751 | 0.0486751 | 4.8817E-07 | 2.0459E-08 | 7.3387E-10 | 2.3012E-11 |
| | 0.50 | 0.0379082 | 0.0379083 | 1.5011E-05 | 1.2652E-06 | 9.1146E-08 | 5.7348E-09 |
| | 0.75 | 0.0295229 | 0.0295244 | 1.0966E-04 | 1.3937E-05 | 1.5121E-06 | 1.4316E-07 |
| | 1.00 | 0.0229925 | 0.0230035 | 4.4503E-04 | 7.5798E-05 | 1.1007E-05 | 1.3936E-06 |
| 0.50 | 0.25 | 0.1947002 | 0.1947002 | 1.9527E-06 | 8.1836E-08 | 2.9355E-09 | 9.2047E-11 |
| | 0.50 | 0.1516327 | 0.1516330 | 6.0043E-05 | 5.0608E-06 | 3.6459E-07 | 2.2939E-08 |
| | 0.75 | 0.1180916 | 0.1180977 | 4.3864E-04 | 5.5750E-05 | 6.0486E-06 | 5.7264E-07 |
| | 1.00 | 0.0919699 | 0.0920139 | 1.7801E-03 | 3.0319E-04 | 4.4029E-05 | 5.5746E-06 |
| 0.75 | 0.25 | 0.4380754 | 0.4380754 | 4.3935E-06 | 1.8413E-07 | 6.6049E-09 | 2.0711E-10 |
| | 0.50 | 0.3411735 | 0.3411743 | 1.3510E-04 | 1.1387E-05 | 8.2032E-07 | 5.1613E-08 |
| | 0.75 | 0.2657062 | 0.2657198 | 9.8693E-04 | 1.2544E-04 | 1.3609E-05 | 1.2884E-06 |
| | 1.00 | 0.2069322 | 0.2070313 | 4.0053E-03 | 6.8219E-04 | 9.9064E-05 | 1.2543E-05 |

$$\mathcal{U}(x,s)=\frac{x^2}{s}+\frac{1}{s^\alpha}\mathcal{L}\left[\frac{\partial^2}{\partial x^2}u\left(\frac{x}{2},\frac{t}{2}\right)\ \frac{\partial}{\partial x}u\left(\frac{x}{2},\frac{t}{2}\right)-\frac{1}{8}\frac{\partial}{\partial x}u\left(x,t\right)-u(x,t)\right] \tag{8.3.16}$$

*The inverse Laplace transform leads to*

$$u(x,t)=x^2+\mathcal{L}^{-1}\left[\frac{x}{s}+\frac{1}{s^\alpha}\mathcal{L}\left\{\frac{\partial^2 u\left(\frac{x}{2},\frac{t}{2}\right)}{\partial x^2}\ \frac{\partial u\left(\frac{x}{2},\frac{t}{2}\right)}{\partial x}-\frac{1}{8}\frac{\partial u\left(x,t\right)}{\partial x}-u(x,t)\right\}\right] \tag{8.3.17}$$

*Homotopy perturbation transform method on Eq.*(8.3.17) *leads to*

$$\begin{aligned}\sum_{n=0}^{\infty}p^n u_n(x,t)&=x^2+p\sum_{n=0}^{\infty}p^n\\ &\mathcal{L}^{-1}\left[\frac{1}{s^\alpha}\mathcal{L}\left[\sum_{k=0}^{n}\frac{\partial^2 u_k\left(\frac{x}{2},\frac{t}{2}\right)}{\partial x^2}\ \frac{\partial u_{n-k}\left(\frac{x}{2},\frac{t}{2}\right)}{\partial x}-\frac{1}{8}\frac{\partial u_n\left(x,t\right)}{\partial x}-u_n(x,t)\right]\right]\end{aligned} \tag{8.3.18}$$

*On comparing the coefficient of like powers of* $p^n$ *on both sides of Equation* (8.3.18)*, we get*

$$\begin{cases}\text{Coefficient of } p^0:\ u_0(x,t)=x^2,\\ \text{Coefficient of } p^{n+1}:\\ u_{n+1}(x,t)=\mathcal{L}^{-1}\left\{\frac{1}{s^\alpha}\mathcal{L}\left(\sum_{k=0}^{n}\frac{\partial^2 u_k(\frac{x}{2},\frac{t}{2})}{\partial x^2}\ \frac{\partial u_{n-k}(\frac{x}{2},\frac{t}{2})}{\partial x}-\frac{1}{8}\frac{\partial u_n(x,t)}{\partial x}-u_n(x,t)\right)\right\},n\geq 0.\end{cases}$$

*The solution of the above recurrence relation yields the following*

$$\begin{aligned}p^0:&u_0(x,t)=x^2,\\ p^1:&u_1(x,t)=\frac{-x^2t^\alpha}{\Gamma(1+\alpha)}\\ p^2:&u_2(x,t)=t^{2\alpha}x\frac{(2^{1-\alpha}+2^2x+1)}{2\Gamma(1+2\alpha)}\\ p^3:&u_3(x,t)=\frac{t^{3\alpha}}{2\Gamma(1+3\alpha)}\left\{-1-2x^2-2^4+2^{-\alpha}+2^{-2\alpha}+2^{-3-\alpha}+2^{-3-2\alpha}\right.\\ &\left.+2^{-2-3\alpha}+2^{-1-2\alpha}x\frac{\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2}\right\}\end{aligned}$$

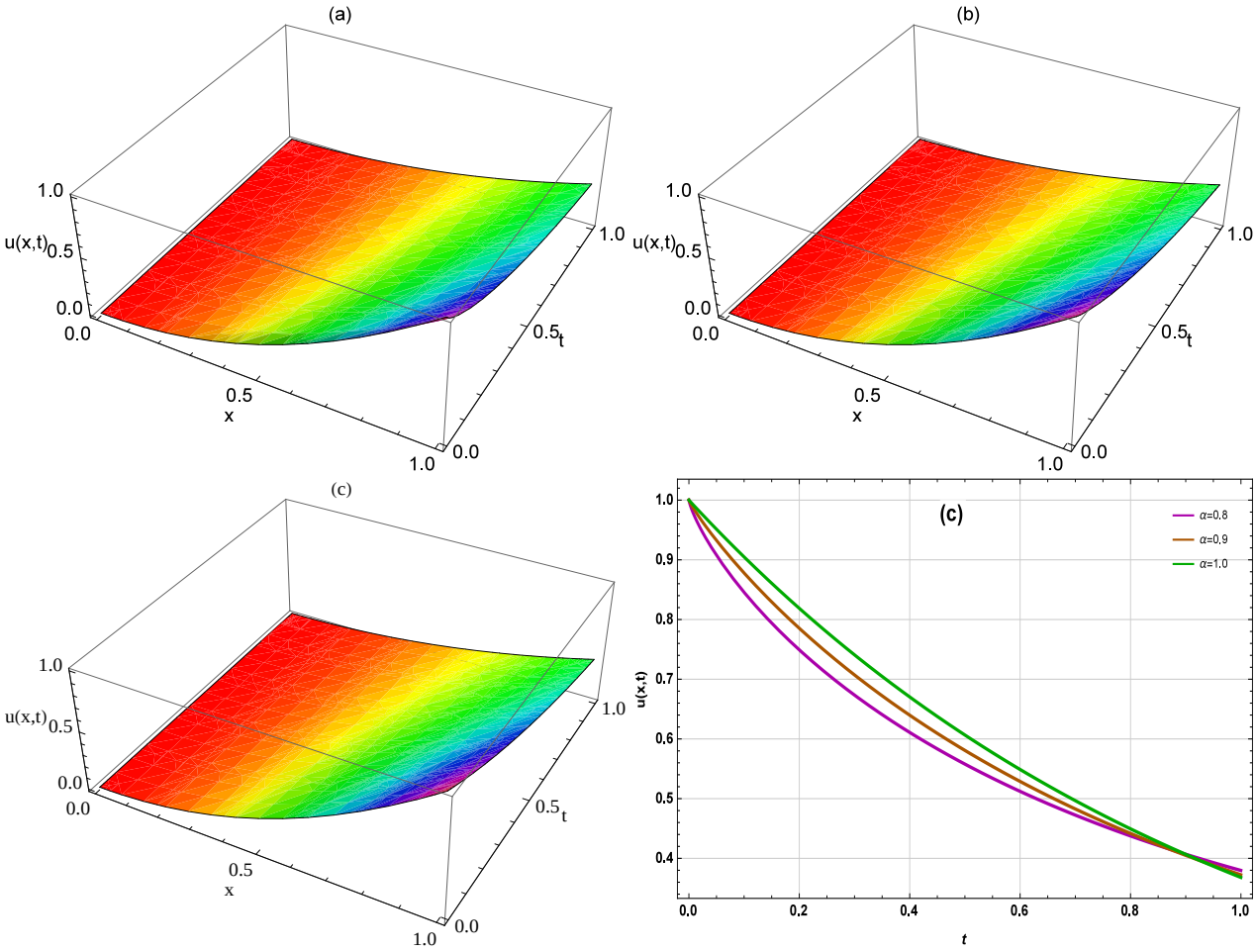


Figure 8.3: The solution behavior of HPTM solution $u$ of Example 8.3.3 for (a) $\alpha = 0.8$; (b) $\alpha = 0.9$; (c) $\alpha = 1.0$; and their plots for different values of $\alpha = 0.8, 0.9, 1.0$ at different time levels $t \in (0, 1)$ and $x = 1$

$$
\begin{aligned}
p^4 : u_4(x,t) = {} & \frac{t^{4\alpha}}{\Gamma(1+4\alpha)} \left\{ (3 \times 2^{-5} - 2^{-3-\alpha} - 2^{-3-2\alpha} + 2^{-3-4\alpha} + 2^{-3-5\alpha}) \right\} \\
& + \frac{t^{4\alpha}}{\Gamma(1+4\alpha)} \left\{ ((-2^{-1-3\alpha} - 2^{-1-2\alpha} - 2^{-1-\alpha} + \frac{3}{4})x + x^2) \right\} \\
& - \frac{t^{4\alpha}}{\Gamma(1+4\alpha)} (-2^{-4-\alpha} + 2^{-5-2\alpha} + 2^{-2-2\alpha} x) \frac{\Gamma(1+2\alpha)}{\Gamma(1+\alpha)^2} \\
& - \frac{t^{4\alpha}}{\Gamma(1+4\alpha)} \left\{ 2^{-4-4\alpha} (-2 + 2^{\alpha} + 2^{3+\alpha} x) \frac{\Gamma(1+3\alpha)}{\Gamma(1+2\alpha) \times \Gamma(1+4\alpha))} \right\}, \ldots
\end{aligned}
$$

*The required solution of Eq.* (8.3.15) *is*

$$u(x,t) = u_0(x,t) + u_1(x,t) + u_2(x,t) + u_3(x,t) + \ldots \tag{8.3.19}$$

*which is a closed form to the exact solution and the solution obtained by Sarkar et al [206].*

*In particular for $\alpha = 1$, the seventh order solution is obtained as*

$$u(x,t) = x^2\left(1 - t + \frac{t^2}{3} - \frac{t^3}{6} + \frac{t^4}{24} - \frac{t^5}{120} + \frac{t^6}{720} - \frac{t^7}{5040} + \ldots\right) \tag{8.3.20}$$

*which is same as obtained by DTM and RDTM [2] , and is a closed form of the exact solution $u(x,t) = x^2\exp(-t)$. The HPTM solutions for $\alpha = 1.0$ are compared with the exact solutions and the absolute errors in different order HPTM solutions are reported in Table 8.3. The findings show that the proposed HPTM results are agreed well with HPM solutions [206], DTM [2] solutions and approaching to the exact solutions. The solution behavior of u for different values of $\alpha = 0.8, 0.9, 1.0$, and the plots of the solutions for $x = 1$ at different time levels $t \leq 1$ are depicted in Figure 8.3.*

## 8.4 Conclusion

In this chapter, the `homotopy perturbation transform method` has been successfully employed for solving the initial value autonomous system of the TFPDEs with proportional delay. Three examples are considered to validate and illustrate the efficiency of the `HPTM`. The proposed solutions agreed excellently with HPM [206], DTM [2] and the two methods AVIM and extended FRDTM reported in Chapter 7. The obtained series solutions converges very fast and does not require any discretization or restrictive conditions. Calculations show that the described HPTM needs a small computation size as compare to HPM [206], DTM [2] and AVIM.

# Bibliography


[1] Abdou MA, Soliman AA, New applications of variational iteration method, Physica D: Nonlinear Phenomena 211 (2005) 1–8.

[2] Abazari R, Ganji M, Extended two-dimensional DTM and its application on nonlinear PDEs with proportional delay, Int. J. Comput. Math. 88(8) (2011) 1749-1762.

[3] Abazari R, Kılıçman A, Application of differential transform method on nonlinear integro–differential equations with proportional delay, Neural Comput & Applic 24 (2014) 391-397.

[4] Abazari R, Borhanifar A, Numerical study of the solution of the Burgers and coupled Burgers equations by a differential transformation method, Comput. Math. Appl. 59 (2010) 2711.

[5] Abazari R, Abazari M, Numerical simulation of generalized Hirota-Satsuma coupled KdV equation by RDTM and comparison with DTM, Commun. Nonlinear. Sci. Numer. Simul. 17 (2012) 619-629.

[6] Abdou MA, Soliman AA, Variational iteration method for solving Burgers' and coupled Burgers' equations, J. Comput. Appl. Math. 181 (2005) 245 - 251.

[7] Adam Y, Highly accurate compact implicit methods and boundary conditions, J. Comput. Phys. 24 (1977) 10-22.

[8] Adomian G, The diffusion Brusselator equation, Comput. Math. Appl. 29(1995) 1-3.

[9] Aksan EN, Quadratic B-spline finite element method for numerical solution of the Burgers equation, Appl. Math. Comput. 174 (2006) 884 - 896.

[10] Akrivis G, Finite difference discretization of the Kuramoto-Sivashinsky equation, Numer. Math. 63 (1992) 1-11.

[11] Arminjon P, Beauchamp C, Numerical solution of Burgers' equations in two-space dimensions, Comput. Methods Appl. Mech. Eng. 19 (1979) 351-365.

[12] Arora G, Joshi V, A computational approach for solution of one dimensional parabolic partial differential equation with application in biological processes, Ain Shams Eng J (2016), http://dx.doi.org/10.1016/j.asej.2016.06.013.

[13] Arora G, Singh BK, Numerical solution of Burgers' equation with modified cubic B-spline differential quadrature method, Applied Math. Comput. 224,(2013) 166-177.

[14] Arora G, Mittal RC, Singh BK, Numerical Solution of BBM-Burger Equation with Quartic B-spline collocation method, Journal of Engineering Science and Technology, Special Issue 12, (2014) 104 - 116.

[15] M. Arshad, D. Lu, J. Wang, (N+1)-dimensional fractional reduced differential transform method for fractional order partial differential equations, Communications in Nonlinear Science and Numerical Simulation 48 (2017) 509–519.

[16] Anders D, Dittmann M, Weinberg K, A higher-order finite element approach to the Kuramoto-Sivashinsky equation, ZAMM Z. Angew. Math. Mech. 92 (2012) 599-607.

[17] Atangana A, Baleanu D, New fractional derivatives with nonlocal and non-singular kernel: Theory and application to heat transfer model, Therm. Sci. 20 (2016) 763-769.

[18] Atangana A, Koca I, Chaos in a simple nonlinear system with Atangana - Baleanu derivatives with fractional order, Chaos, Solitons Fractals 89 (2016) 447-454.

[19] Ayaz F, On the two-dimensional differential transform method, Applied Mathematics and Computation 143 (2003) 361–374.

[20] Ayati Z, Biazar J, On convergence of homotopy perturbation method, Journal of Egyptian Mathematical Society 23 (2015) 424-428.

[21] Bakewell P, Lumley JL, Viscous sublayer and adjacent wall region in turbulent pipe flow, Phys. Fluids 10, 1967., 1880-1889.

[22] Baleanu D, Machado JAT, Luo AC, *Fractional dynamics and control*, Springer, New York, 2012.

[23] Bastani M, Salkuyeh DK, A highly accurate method to solve Fisher's equation, Pramana- J. Phys. Indian Academy of Sciences. 78(3) ( 2012) 335-346.

[24] Bashan A, Karakoc SBG, Geyikli T, Approximation of the KdVB equation by the quintic B-spline differential quadrature method, Kuwait J. Sci. 42 (2) (2015) 67 - 92.

[25] Batchelor GK, An Introduction to Fluid Dynamics, Cambridge Univ. Press (2002).

[26] Bateman H, Some recent researches on the motion of fluids, Mon. Weather Rev. 43 (1915) 163 - 170

[27] Bellman R, Kashef BG, Casti J, Differential quadrature: a technique for the rapid solution of nonlinear differential equations, J. Comput. Phys. 10 (1972) 40 - 52.

[28] Bellman R, Kashef B, Lee ES, Vasudevan R., Solving hard problems by easy methods: differential and integral quadrature, Comput. Math. Appl. 1 (1) (1975) 133-143.

[29] Bert CW, Malik M, Differential quadrature method in computational mechanics, Appl. Mech. Rev., 49 (1) (1996) 1-28.

[30] Bhatt HP, Khaliq AQM, Fourth-order compact schemes for the numerical simulation of coupled Burgers equation, Comput. Phys. Commun. 200 (2016) 117 - 138.

[31] Biazar J, Aminikhah H, Exact and numerical solutions for non-linear Burgers' equation by VIM, Math. Comput. Modelling 49 (2009) 1394-1400.

[32] Biazar J, Ghanbari B, The homotopy perturbation method for solving neutral functional-differential equations with proportional delays, J. King Saud Univ. Sci. 24 (2012) 33-37.

[33] Biazar J, Ghazvini H, Convergence of the homotopy perturbation method for partial differential equations, Nonlinear Analysis: Real World Applications 10 (2009) 2633-2640.

[34] Birajdar GA, Numerical solution of time fractional Navier-Stokes equation by discrete adomaian decomposition method, Nonlinear Engineering 3 (1) (2014) 21 - 26.

[35] Burgers JM, Mathematical example illustrating relations occurring in the theory of turbulent fluid motion, Trans. Roy. Neth. Acad. Sci. Amsterdam 17 (1939) 1 - 53.

[36] Burgers JM, A mathematical model illustrating the theory of turbulence, Adv. Appl. Mech, vol. 1, Academic Press, New York, 1948, pp. 171 - 199

[37] Burgers JM, The Non-Linear Diffusion Equation: Asymptotic Solutions and Statistical Problems, New York, Springer, 1974.

[38] Campos MD, Romão EC, A high-order finite-Difference scheme with a linearization technique for solving of three-dimensional Burgers equation, Comput. Modelling in Engg. Sci. 103(3) (2014) 139-154.

[39] Caputo M, Mainardi F, Linear models of dissipation in anelastic solids. Rivista del Nuovo Cimento 1 (1971) 161 - 98.

[40] Carpinteri A, Mainardi F, *Fractals and fractional calculus in continuum mechanics*, Springer Verlag, Wien, New York, 1997.

[41] Chaurasia VBL, Kumar D, Solution of the time-fractional Navier-Stokes equation, Gen. Math. Notes 4(2) (2011) 49-59.

[42] Chen CK, Ho SH, Solving partial differential equations by two dimensional differential transform, Appl. Math. Comput. 106 (1999) 171–179.

[43] Cheong C, Lee S Grid-optimized dispersion-relation-preserving schemes on general geometries for computational aeroacoustics, J. Comput. Phys. 174 (2001) 248-276.

[44] Chen X, Wang L, The variational iteration method for solving a neutral functional-differential equation with proportional delays, Comput. Math. Appl. 59 (2010) 2696-2702.

[45] Chen W, Zhong TX, Shu C, A Lyapunov formulation for efficient solution of the Poisson and convection-diffusion equations by differential quadrature method, J. Comput. Phys. 141 (1) (1998) 78-84.

[46] Chu PC, Fan C, A three-point combined compact difference scheme, J. Comput. Phys. 140 (1998) 370-399.

[47] Chu PC, Fan C, A three point sixth order non uniform combined compact difference scheme, J. Comput. Phys.148, (1999) 663–674.

[48] Chun C, Fourier-series-based variational iteration method for a reliable treatment of heat equations with variable coefficients, International Journal of Nonlinear Sciences and Numerical Simulation 10 (2009) 1383-1388.

[49] Chung J, Kim E, Kim Y, Asymptotic agreement of moments and higher order contraction in the Burgers equation, J. Diff. Equ. 248 (10) (2010) 2417 - 2434.

[50] Civan F, Sliepcevich CM, Differential quadrature for multi-dimensional problems, J. Math. Anal. Appl., 101 (1984) 423-443

[51] Civalek Ö, Harmonic differential quadrature finite differences coupled approaches for geometrically nonlinear static and dynamic analysis of rectangular plates on elastic foundation, J. Sound Vib. 294 (2006) 966-80.

[52] Cole JD, On a quasilinear parabolic equations occurring in aerodynamics, Quart. Appl. Math. 9 (1951) 225 - 236.

[53] Collatz L, The Numerical Treatment of Differential Equations, Springer Verlag, New York (1966) pp.538

[54] Conte R, Exact solutions of nonlinear partial differential equations by singularity analysis, Lecture notes in Physics, Springer 632 (2003) 1-83.

[55] Dağ İ, Irk D, Sahin A, B-Spline collocation methods for numerical solutions of the Burgers' equation, Math. Probl. Eng. 5 (2005) 521-538.

[56] Dehghan M, Taleei A, A compact split-step finite difference method for solving the nonlinear Schrodinger equations with constant and variable coefficients. Comp. Phys. Commun. 181(1) (2010) 43-51.

[57] Deghan M, Hamidi A, Shakourifar M, The solution of coupled Burgers' equations using Adomian-Pade technique, Appl. Math. Comput. 189 (2007) 1034-47.

[58] Deng D, Pan T, A fourth-order singly diagonally implicit Runge-Kutta method for solving one-dimensional Burgers' equation, IAENG International Journal of Applied Mathematics, 45 (4), (2015) 327 - 333.

[59] Drain PG, Johnson RS, Solitons: An Introduction. Cambridge University Press, New York (1989).

[60] El-Shahed M, Salem A, On the generalized Navier-Stokes equations, Appl. Math. Comput. 156(1) (2005) 287 - 293.

[61] Ersoy O, Dag I, The extended B-spline collocation method for numerical solutions of Fisher equation, AIP Conf. Proc. 1648 (2015) 370011, http://dx.doi.org/10.1063/1.4912600.

[62] Esipov SE, Coupled Burgers equations: a model of polydispersive sedimentation, Phys. Rev. 52 (1995) 3711 - 3718.

[63] Faraz N, Khan Y, Yildirim A, Analytical approach to two–dimensional viscous flow with a shrinking sheet via variational iteration algorithm-II, Journal of King Saud University (2010), doi:10.1016/j.jksus.2010.06.010.

[64] Fan EG, Zhang HQ, A note on the homogeneous balance method. Phys. Lett. A 264, (1998) 403-406.

[65] Fletcher CAJ, Generating exact solutions of the two dimensional Burgers equation, Int. J. Numer. Meth. Fluids 3 (1983) 213 - 216.

[66] Gamet L, Ducros F, Nicoud F, Poinsot T, Compact finite difference schemes on non-uniform meshes. Application to direct numerical simulation of compressible flows, Int. J. Numer. Meth. Fluids 29 (1999) 159-191.

[67] Ganji DD, The application of He's homotopy perturbation method to nonlinear equations arising in heat transfer, Phys Lett A 355 (2006) 337-341.

[68] Ganji ZZ, Ganji DD, Ganji AD, Rostamian M, Analytical solution of time-fractional Navier-Stokes equation in polar coordinate by homotopy perturbation method, Numerical Methods for Partial Differential Equations 26 (1) (2010) 117-124.

[69] Ganjefar S, Rezaei S, Modified homotopy perturbation method for optimal control problems using the Padè approximant, Applied Mathematical Modelling (2016), http://dx.doi.org/10.1016/j.apm.2016.02.039.

[70] Goldfain E, Fractional dynamics, cantorian space-time and the gauge hierarchy problem. *Chaos, Solitons and Fractals* **22(3)** (2004) 513-520.

[71] Geng F, Lina Y, Cui M, A piecewise variational iteration method for Riccati differential equations, Computers and Mathematics with Applications 58 (2009) 2518 - 2522.

[72] Geodheer WJ, Potters JHHM, A compact finite difference scheme on a non-equidistant mesh, J. Comput. Phys. 61 (1985) 269-279.

[73] Gomez H, París J, Numerical simulation of asymptotic states of the damped Kuramoto-Sivashinsky equation, Phys. Rev. E 93 (2011) 046702.

[74] Gondal MA, Khan M, Omrani K, A new analytical approach to two dimensional magneto-hydrodynamics flow over a nonlinear porous stretching sheet by Laplace Padè decomposition method, Results Math. 63 (2013) 289 - 301.

[75] Goufo EFD, Chaotic processes using the two-parameter derivative with non-singular and non-local kernel: Basic theory and applications, Chaos: An Interdisciplinary Journal of Nonlinear Science, http://dx.doi.org/10.1063/1.4958921

[76] Goufo EFD, Stability and convergence analysis of a variable order replicator-mutator process in a moving medium, Journal of Theoretical Biology 403 (2016) 178-187.

[77] Gottlieb S, Ketcheson DI, Shu CW, High order strong stability preserving time discretizations, J. Sci. Comput. 38 (2009) 251 - 289.

[78] Guckenheimer J, Strange attractors in fluids: Another view, Ann. Rev. Fluid Mech. 18 (1986) 15-32.

[79] Gupta PK, Approximate analytical solutions of fractional Benney-Lin equation by reduced differential transform method and the homotopy perturbation method, Comp. Math. Appl. 58 (2011) 2829-2842

[80] Gürarslan G, Karahan H, Alkaya D, Sari M, Yasar M, Numerical solution of advection-diffusion equation using a sixth-order compact finite difference method. Math. Prob. Engg. Volume 2013 (2013), Article ID 672936, 7 pages http://dx.doi.org/10.1155/2013/672936.

[81] Hassanien IA, Salama AA, Hosham HA, Fourth-order finite difference method for solving Burgers equation, Appl. Math. Comput. 170 (2005) 781 - 800.

[82] He JH, A tutorial review on fractal space time and fractional calculus, Int. J. Theor. Phys. 53 (11) (2014) 3698- 3718.

[83] He JH, A new approach to nonlinear partial differential equations, Communications in Nonlinear Science and Numerical Simulation 2(4) (1997) 230-235.

[84] He JH, *Nonlinear oscillation with fractional derivative and its applications*, Int. Conf. Vibrating Engg', Dalian, (1998) 288-91.

[85] He JH, Approximate analytical solution for seepage flow with fractional derivatives in porous media. *Comput. Methods Appl. Mech. Eng.* **167** (1998) 57-68

[86] He JH, Homotopy perturbation technique, *Comput. Methods Appl. Mechanics Engg.* **178** (1999) 257-62.

[87] He JH, Homotopy perturbation method: a new nonlinear analytical technique, Applied Mathematics and Computation 135 (2003) 73–79.

[88] He JH, Comparison of homotopy perturbation method and homotopy analysis method, Applied Mathematics and Computation 156 (2004) 527–539.

[89] He JH, The homotopy perturbation method for nonlinear oscillators with discontinuities, Applied Mathematics and Computation 151 (2004) 287–292.

[90] He JH, Homotopy perturbation method for bifurcation of nonlinear problems, International Journal of Nonlinear Sciences and Numerical Simulation 6 (2005) 207-208.

[91] Hesameddini E, Latifizadeh H, A new vision of the He's homotopy perturbation method, International Journal of Nonlinear Sciences and Numerical Simulation 10 (2009) 1415-1424.

[92] He JH, Variational iteration methoda kind of nonlinear analytical technique: some examples, International Journal of Nonlinear Mechanics 34 (1999) 699–708.

[93] He JH, Wu XH, Variational iteration method: new development and applications, Computers & Mathematics with Applications 54 (2007) 881–894.

[94] Hesch C, Schu**B** S, Dittmann M, Franke M, Weinberg K, Isogeometric analysis and hierarchical refinement for higher-order phase-field models. Comput. Methods Appl. Mech. Engrg. 303 (2016) 185-207.

[95] Hirsh RS, Higher order accurate difference solutions of fluid mechanics problems by a compact differencing technique, J. Comput. Phys. 19 (1975) 90–109.

[96] Hilfer R, *Applications of fractional calculus in physics*, World scientific, Singapore, 2000.

[97] Hooper AP, Grimshaw R, Nonlinear instability at the interface between two viscous fluids, Phy. Fluids 28 (1985) 37-45.

[98] Hongqing Z, Huazhong S, Meiyu D, Numerical solutions of two-dimensional Burgers' equations by discrete Adomian decomposition method, Comp. Math. App. 60 (2010) 840-848.

[99] Ishteva MK, Properties and applications of the Caputo fractional operator, Master thesis, 2005.

[100] Jaber KK, Ahmad RS, Analytical solution of the time fractional Navier-Stokes equation (2017), http://dx.doi.org/10.1016/j.asej.2016.08.021

[101] Jain MK, Numerical Solution of Differential Equations, second ed., Wiley, New York, NY, 1983.

[102] Jain PC, Holla DN, Numerical solution of coupled Burgers' equations, Int. J. Numer. Meth. Eng. 12, 213 (1978).

[103] Johnston SJ, Jafari H, Moshokoa SP, Ariyan VM, Baleanu D, Laplace homotopy perturbation method for Burgers equation with space and time fractional order, Open Physics 14(1) (2016) 247-252.

[104] Kadalbajoo MK, Sharma KK, Awasthi A, A parameter-uniform implicit difference scheme for solving time dependent Burgers' equation, Appl. Math. Comput. 170 (2005) 1365-1393.

[105] Kashuri A, Fundo A, A new integral transform, Adv. Theoretical Appl. Math 8(1) (2013) 27-43.

[106] Kashuri A, Fundo A, Kreku M, Mixture of a new integral transform and homotopy perturbation method for solving nonlinear partial differential equations, Adv Pure Math. 3(2013) 317-23.

[107] Kaya D, An explicit solution of coupled viscous Burgers' equation by the decomposition method, IJMMS 27(11) (2001) 675-80.

[108] Kelleci A, Yıldırım A, An efficient numerical method for solving coupled Burgers' equation by combining homotopy perturbation and Pade techniques, Numerical Methods for Partial Differential Equations, http://dx.doi.org/10.1002/num.20565.

[109] Keller AA, Contribution of the delay differential equations to the complex economic macrodynamics, WSEAS Transactions on Systems 9(4) (2010) 258-271.

[110] Keskin Y, Oturanc G, Reduced differential transform method for partial differential equations, Int. J. Nonlinear. Sci. Numer. Simul. 10(6) (2009) 741-749.

[111] Keskin Y, Oturanc G, Reduced differential transform method: a new approach to factional partial differential equations, Nonlinear Sci. Lett. A 1 (2010) 61-72.

[112] Khader MM, Kumar S, Abbasbandy S, New homotopy analysis transform method for solving the discontinued problems arising in nanotechnology, Chin. Phys. B 22 (11) (2013) 110201.

[113] Khater AH, Temsah RS, Numerical solutions of the generalized Kuramoto-Sivashinsky equation by Chebyshev spectral collocation methods, Comput. Math. Appl. 56, (2008) 1465-1472.

[114] Khater AH, Temsah RS, Hassan MM, A Chebyshev spectral collocation method for solving Burgers-type equations, J Comput. Appl. Math. 222(2) (2008) 333-50.

[115] Khan M, Hussain M, Application of Laplace decomposition method on semi-infinite domain, Numerical Algorithms 56 (2011) 211-218.

[116] Khan M, Gondal MA, A reliable treatment of Abel's second kind singular integral equations, Appl. Math. Lett. 25 (11) (2012) 1666-1670.

[117] Khan M, Gondal MA, Batool SI, A new modified Laplace decomposition method for higher order boundary value problems, Comput. Math. Organ. Theory 19(4) (2013) 446-459.

[118] Khan M, Gondal MA, Kumar S, A new analytical solution procedure for nonlinear integral equations, Mathematical and Computer Modelling 55 (2012) 1892 - 1897.

[119] Khan M, Gondal MA, Hussain I, Vanani SK, A new comparative study between homotopy analysis transform method and homotopy perturbation transform method on semi-infinite domain, Math. Comput. Model. 55 (2012) 1143-1150.

[120] Kilbas AA, Srivastava HM, Trujillo JJ, Theory and applications of fractional differential equations, Elsevier, Amsterdam (2006).

[121] Klafter J, Lim SC, Metzler R, *Fractional Dynamics: Recent Advances*, World Scientific, 2012.

[122] Kopal Z, Numerical Analysis,Wiley, New York (1961) pp. 552.

[123] Korkmaz A, Numerical solutions of some nonlinear partial differential equations using differential quadrature method (Thesis of Master Degree), Eskişehir Osmangazi University, 2006.

[124] Korkmaz A, Numerical solutions of some one dimensional partial differential equations using B-spline differential quadrature method (Doctoral Dissertation), Eskişehir Osmangazi University, 2010.

[125] Korkmaz A, Numerical algorithms for solutions of Kortewegde Vries equation, Numer. Methods Partial Diff. Equ. 26 (6) (2010) 1504-1521.

[126] Korkmaz A, Akmaz HK, Numerical simulations for transport of conservative pollutants, Selcuk J. Appl. Math. 16 (1) (2015).

[127] Korkmaz A, Akmaz HK, Extended B-spline differential quadrature method for nonlinear viscous Burgers equation, in: Proceedings of International Conference on Mathematics and Mathematics Education, Elazig , Turkey 1214 May, 2016, 2016, p. 323.

[128] Korkmaz A, Dağ İ, Cubic B-spline differential quadrature methods and stability for Burgers equation, Eng. Comput. Int. J. Comput. Aided Eng. Software 30 (3) (2013) 320-344.

[129] Korkmaz A, Dağ İ, Numerical simulations of boundary-forced RLW equation with cubic B-spline-based differential quadrature methods, Arab. J. Sci. Eng. 38 (2013) 1151 - 1160.

[130] Korkmaz A, Dağ İ, Quartic and quintic B-spline methods for advection diffusion equation, Appl. Math. Comput. 274 (2016) 208 - 219.

[131] Korkmaz A, Dağ İ, Polynomial based differential quadrature method for numerical solution of nonlinear Burgers equation, J. Franklin Inst. 348 (10) (2011) 2863-2875.

[132] Korkmaz A, Dağ İ, Cubic B-spline differential quadrature methods for the advection-diffusion equation, Int. J. Numeri. Methods for Heat and Fluid Flow 22 (8) (2012) 1021-1036.

[133] Korkmaz A, Aksoy AM, Dağ İ, Quartic B-spline differential quadrature method, Int. J. Nonlinear Sci. 11 (4), (2011) 403-411.

[134] Korkmaz A, Dağ İ,Shock wave simulations using sinc differential quadrature method, Engg. Comput. Int. J. Computer-Aided Engg. Soft. 28(6), (2011) 654-674.

[135] Kubatko JE, Yeager BA, Ketcheson DI, Optimal strong-stability-preserving RungeKutta time discretizations for discontinuous Galerkin methods, ¡http:// www.davidketcheson.info/assets/papers/dg-ssp-stability.pdf.

[136] Kumar S, Kumar D, Abbasbandy S, Rashidi MM, Analytical solution of fractional Navier-Stokes equation by using modified Laplace decomposition method, Ain Shams Engineering Journal 5(2)(2014) 569-574.

[137] Kumar S, Kumar D, Fractional modelling for BBM-Burger equation by using new homotopy analysis transform method, J. Associ. of Arab Uni. for Basic and App. Sci. 16 (2014) 16-20.

[138] Kumar D, Singh J, Kumar S, Numerical computation of nonlinear fractional Zakharov- Kuznetsov equation arising in ion-acoustic waves, J. Egypt. Math. Society 22 (3) (2014) 373-378.

[139] Kumar D, Singh J, Kumar S, A fractional model of Navier-Stokes equation arising in unsteady flow of a viscous fluid, J. Associ. of Arab Uni. for Basic and App. Sci. 17 (2015) 14-19.

[140] Kumar D, Singh J, Baleanu D, Numerical computation of a fractional model of differential-difference equation, J. of Compu. and Nonlinear Dynamics, 11(6) (2016) 061004.

[141] Kumar D, Singh J, Baleanu D, A hybrid computational approach for Klein-Gordon equations on Cantor sets, Nonlinear Dynamics 87(1) (2017) 511-517

[142] Kumar M, Pandit S, A composite numerical scheme for the numerical simulation of coupled Burgers equation, Computer Physics Communications 185 (2014) 809-817.

[143] Kumar S, Kumar A, Baleanu D, Two analytical methods for time fractional nonlinear coupled Boussinesq Burger's equations arise in propagation of shallow water waves, Nonlinear Dynamics 85 (2) (2016) 699-715.

[144] Kuramoto Y, Tsuzuki T, Persistent propagation of concentration waves in dissipative media far from thermal equilibrium, Prog. Theor. Phys. 55 (1976) 356-569.

[145] Khuri SA, A Laplace decomposition algorithm applied to a class of nonlinear differential equations, Journal of Applied Mathematics 1 (2001) 141–155.

[146] Kurulay M, Secer A, Akinlar MA, A new approximate analytical solution of Kuramoto-Sivashinsky equation using Homotopy analysis method, Appl. Math. Inf. Sci. 7(1), ( 2013) 267-271.

[147] Lai H, Ma C, A new lattice Boltzmann model for solving the coupled viscous Burgers equation, Physica A 395 (2014) 445-457

[148] Lai H, Ma C, Lattice Boltzmann method for the generalized KuramotoSivashinsky equation, Physica A 388 (2009) 1405-1412.

[149] Lakestania M, Dehghan M, Numerical solutions of the generalized Kuramoto-Sivashinsky equation using B-spline functions. Appl. Math. Model. 36(2) ( 2012) 605-617.

[150] Lan H, Wang K, Exact solutions for two nonlinear equations, J. Phy. A, Math. Gen. 23 (1990) 3923-3928.

[151] Lee W, Tridiagonal matrices: Thomas algorithm, Scientific Computation, University of Limerick. ¡http://www3.ul.ie/wlee/ms6021thomas.pdf

[152] Lele SK, Compact finite difference schemes with spectral-like resolution, J. Comput. Phy. 103(1), (1992) 16-42.

[153] Liao W, An implicit fourth-order compact finite difference scheme for one-dimensional Burgers equation, Appl. Math. Comput. 206 (2008) 755 - 764.

[154] Lin S, Wang C, Dai Z, New exact traveling and non-traveling wave solutions for (2+1) dimensional Burgers equation, Appl. Math. Comput. 216(10) (2010) 3105-3110.

[155] Liu J, Hou G, Numerical solutions of the space-and time-fractional coupled burgers equations by generalized differential transform method, Appl. Math. Comput. 217 (16) (2011) 7001-7008 .

[156] Logan JD, An Introduction to Non - Linear Partial Differential Equations, New York: Wiley (1994).

[157] Malekzadeh P, Karami G, Polynomial and harmonic differential quadrature methods for free vibration of variable thickness thick skew plates, Eng. Struc., 27 (2005) 1563-1574.

[158] Malekzadeh P, Karami G, A mixed differential quadrature and finite element free vibration and buckling analysis of thick beams on two parameter elastic foundations, Appl. Math. Modelling, 32 (2008) 1381-1394.

[159] Madani M, Fathizadeh M, Homotopy perturbation algorithm using Laplace transformation, Nonlinear Science Letters A 1 (2010) 263–267.

[160] Madani M, Fathizadeh M, Khan Y, Yildirim A, On the coupling of the homotopy perturbation method and Laplace transformation, Mathematical and Computer Modelling 53 (2011) 1937-1945.

[161] Madaki G, Abdulhameed M, Ali M, Roslan R, Solution of the Falkner-Skan wedge flow by a revised optimal homotopy asymptotic method, Madaki et al. SpringerPlus (2016) 5-513.

[162] Magesh N, Saravanan A, The reduced differential transform method for solving the systems of two dimensional nonlinear Volterra integro-differential equations, Proceedings of the International conference on mathematical sciences (ICMS-2014), Elsevier, ISBN-978-93-5107-261-4, 217–220.

[163] McDonough JM, Lectures in Elementry fluid Dynamics, Physics, Mathematics and Applications, (2009).

[164] Mead J, Zubik-Kowal B, An iterated pseudospectral method for delay partial differential equations, Appl. Numer. Math. 55 (2005) 227-250.

[165] Mehra M, Patel KS, Algorithm 986: a suite of compact finite difference schemes, ACM Trans. Math. Softw. 44, 2, Article 23 (October 2017), 31 pages. https://doi.org/10.1145/3119905

[166] Miller KS, Ross B, *An Introduction to the fractional calculus and fractional differential equations*, Wiley, New York, 1993.

[167] Mittal RC, Arora G, Quintic B-spline collocation method for numerical solution of the Kuramoto-Sivashinsky equation, Commun Nonlinear Sci Numer Simulat 15, (2010) 2798-2808

[168] Mittal RC, Arora G, Numerical solution of the coupled viscous Burgers equation, Commun. Nonlinear Sci. Numer. Simul. 16 (2011) 1304 - 1313.

[169] Mittal RC, Bhatia R, A numerical study of two dimensional hyperbolic telegraph equation by modified B-spline differential quadrature method Appl. Math. Comput. 244 (2014) 976-997.

[170] Mittal RC, Jiwari R, Differential quadrature method for numerical solution of coupled viscous Burgers' equations, Int. J. Comput. Methods Engg. Sci. Mechanics, 13 (2) (2012) 88-92.

[171] Mittal RC, Jain RK, Numerical solutions of nonlinear Burgers' equation with modified cubic B-splines collocation method, Appl. Math. Comput. 218 (2012) 7839–7855.

[172] Mittal RC, Tripathi A, A colocation method for numerical solutions of coupled Burgers' equations, Int. J. Comput. Methods Engg. Sci. Mechanics 15(4) (2014) 457-471.

[173] Mohamed AR, Al-luhaibi Mohamed S, Application of Sumudu decomposition method for solving Nonlinear Wave-like equations with variable coefficients, Electr J Math Anal Appl 4(1)(2016) 116-24.

[174] Mohanty RK, Dai W, Han F, Compact operator method of accuracy two in time and four in space for the numerical solution of coupled viscous Burgers equations, Appl. Math. Comput. 256 (2015) 381 - 393.

[175] Mohyud-Din ST, Yildirim A, Homotopy perturbation method for advection problems, Nonlinear Science Letters A 1 (2010) 307-312.

[176] Mokhtari R, Toodar AS, Chegini NG, Application of the generalized differential quadrature method in solving Burgers' equations, Commun. Theor. Phys. 56(6) (2011) 1009-1015.

[177] Momani S, Odibat Z, Analytical solution of a time-fractional Navier-Stokes equation by adomaian decomposition method, Appl. Math. Comput. 177 (2006) 488 - 494.

[178] Momani S, Odibat Z, Homotopy perturbation method for nonlinear partial differential equations of fractional order, Physics Letters A 365 (2007) 345-350.

[179] Momani S, Erjaee GH, Alnasr MH, The modified homotopy perturbation method for solving strongly nonlinear oscillators, Computers and Mathematics with Applications 58 (2009) 2209-2220.

[180] Navier CLMH, Memoire sur les lois du mouvement des fluides, Mem. Acad. Sci. Inst. France, 6 (1822) 389-440.

[181] Nazari D, Shahmorad S, Application of the fractional differential transform method to fractional - order integro - differential equations with nonlocal boundary conditions, Journal of Computational and Applied Mathematics 234 (2010) 883–891.

[182] Nee J, Duan J, Limit set of trajectories of the coupled viscous Burgers' equations, Appl Math Lett 11(1) (1998) 57-61.

[183] Odibat ZM, Shawagfeh N, Generalized Taylor's formula, Appl. Math. Comput. 186 (2007) 286 - 293.

[184] Odibat ZM, Differential transform method for solving Volterra integral equations with separable kernels, Math. Comput. Modelling 48 (2008) 1141–1149.

[185] Odibata Z, Momani S, A generalized differential transform method for linear partial differential equations of fractional order, Applied Mathematics Letters 21 (2008) 194–199.

[186] Odibat ZM, Momani S, Application of varitional iteration method to nonlinear differential equations of fractional order, Int J Nonlinear Sci Numeri Simulation 7(1) (2006) 27-34.

[187] Odibat Z, Momani S, A reliable treatment of homotopy perturbation method for Klein-Gordon equations, Phys Lett A 365 (2007) 351-357.

[188] Odibat Z, Momani S, The variational iteration method: an efficient Scheme for handling fractional partial differential equation in fluid mechanics, Comput. Math. Appl. 58 (2009) 2199 - 2208.

[189] Odibat ZM, A study on the convergence of variational iteration method, Math. Comput. Model. 51 (2010) 1181-1192.

[190] Ohkitani K, Dowker M, Numerical study on comparison of Navier-Stokes and Burgers equations. Physics of Fluids, 24 (5). (2012) 055113.

[191] Edmundo Capelas de Oliveira, José António Tenreiro Machado, A review of definitions for fractional derivatives and integral, Mathematical Problems in Engineering, Volume 2014,(ID 238459) 6 pages, http://dx.doi.org/10.1155/2014/238459

[192] Ozis T, Esen A, Kutluay S, Numerical solution of Burgers' equation by quadratic B-spline finite elements, Appl. Math. Comput. 165 (2005) 237–249.

[193] Prakash A, Kumar M, Sharma KK, Numerical method for solving fractional coupled Burgers equations, Appli. Math. Comput. 260 (2015) 314-320.

[194] Podlubny I, Fractional differential equations, Academic Press, San Diego, 1999.

[195] Polyanin AD, Zhurov AI, Functional constraints method for constructing exact solutions to delay reaction-diffusion equations and more complex nonlinear equations, Commun. Nonlinear Sci. Numer. Simul. 19 (3) (2014) 417-430.

[196] Quan JR, Chang CT, New insights in solving distributed system equations by the quadrature methods-I, Comput. Chem. Eng. 13 (1989) 779–788.

[197] Quan JR, Chang CT, New insights in solving distributed system equations by the quadrature methods-II, Comput. Chem. Eng. 13 (1989) 10171024.

[198] Rademacher J, Wattenberg R, Viscous shocks in the destabilized Kuramoto-Sivashinsky equation. J Comput Nonlinear Dyn 1 (2006) 336-347.

[199] Rafei M, Ganji DD, Explicit solutions of helmhotz equation and fifth-order KdV equation using homotopy perturbation method, International Journal of Nonlinear Sciences and Numerical Simulation 7 (2006) 321-328.

[200] Ragab AA, Hemida KM, Mohamed MS, Abd El Salam MA, Solution of time-fractional Navier-Stokes equation by using homotopy analysis method, Gen. Math. Notes 13 (2) (2012) 13-21.

[201] Rajaee M, Karlsson SKF and Sirovich L, Low-dimensional description of free shear flow coherent structures and their dynamical behavior, J. Fluid Mech. 258 (1994) 1–20.

[202] Rashid A, Ismail AI Md, A fourier pseudospectral method for solving coupled viscous Burgers' equations, Comput. Methods Appl. Math. 9 (2009) 412420.

[203] Ray SS, Sahoo S, A comparative study on the analytic solutions of fractional coupled sine-Gordon equations by using two reliable methods, Appli. Math. Comput. 253 (2015) 72-82.

[204] Ray SS, A new coupled fractional reduced differential transform method for the numerical solutions of 2 dimensional time fractional coupled Burger equations, Modelling and Simulation in Engineering, Volume 2014 (2014), Article ID 960241, 12 pages.

[205] Ravi Kanth ASV, Aruna K, Differential transform method for solving linear and non-linear systems of partial differential equations, Physics Letters A 372 (2008) 6896–6898.

[206] Sakar MG, Uludag F, Erdogan F, Numerical solution of time-fractional nonlinear PDEs with proportional delays by homotopy perturbation method, Appl. Math. Modelling (2016), http://dx.doi.org/10.1016/j.apm.2016.02.005.

[207] Sakar MG, Ergören H, Alternative variational iteration method for solving the time-fractional Fornberg - Whitham equation, Appl. Math. Model. 39 (14) (2015) 3972 - 3979.

[208] Sakar MG, Erdogan F, The homotopy analysis method for solving the time-fractional Fornberg-Whitham equation and comparison with Adomian's decomposition method, Appl. Math. Model. 37 (20-21) (2013) 1634 - 1641.

[209] Saka B, Sahin A, Dag I, B-spline collocation algorithms for numerical solutions of the RLW equation, Numer. Methods Partial Diff. Equ. 27 (3) (2009) 581607.

[210] Salas A.H., Symbolic computation of solutions for a forced Burgers equation, Appl. Math. Comput. 216 (1) (2010) 1826.

[211] Saravanan A, Magesh N, A comparison between the reduced differential transform method and the Adomian decomposition method for the Newell - Whitehead -Segel equation, J. Egyptian Math. Soc., 21 (3) (2013) 259-265.

[212] Saravanan A, Magesh N, An efficient computational technique for solving the Fokker-Planck equation with space and time fractional derivatives, Journal of King Saud University Science 28 (2016) 160 - 166

[213] Sanyasiraju, YVSS, Manjula V, Higher order semi-compact scheme to solve transient incompressible Navier-Stokes equations, Comput. Mech. 35(6) (2005) 441-448.

[214] Sayevand K, Pichaghchi K, Analysis of nonlinear fractional KdV equation based on He's fractional derivative, Nonlinear Sci. Lett. A, 7 (3) (2016) 77-85.

[215] Sari M, Gürarslan G, A sixth-order compact finite difference scheme to the numerical solutions of Burgers' equation, Appl. Math. Comput. 208(2) (2009) 475-483.

[216] Sari M, Gürarslan G, A sixth-order compact finite difference method for the one-dimensional sine-Gordon equation, Int. J. Numer. Meth. Bio. Engg. 27(7) (2011) 1126-1138.

[217] Sari M, Gürarslan G, Dăg I, A compact finite difference method for the solution of the generalized Burgers-Fisher equation, Numer. Methods for Partial Differential Equations 26(1) (2010) 125-134.

[218] Saha K, Twinkle S, Kilicman A, Combination of integral and projected differential transform methods for time-fractional gas dynamics equations, Ain Shams Eng J 2017, http://dx.doi.org/10.1016/j.asej.2016.09.012

[219] Shah K, Singh T, A Solution of the Burger's equation arising in the longitudinal dispersion phenomenon in fluid flow through porous media by mixture of new integral

transform and homotopy perturbation method, J Geosci Environ Protect 2015;3:24-30. http://dx.doi.org/10.4236/gep.2015.34004.

[220] Shah K, Singh T, The mixture of new integral transform and homotopy perturbation method for solving discontinued problems arising in nanotechnology, Open J Appl Sci 2015;5:68895. doi: http://dx.doi.org/ 10.4236/ojapps.2015.511068.

[221] Shakeri F, Dehghan M, Solution of delay differential equations via a homotopy perturbation method, Mathematical and Computer Modelling 48 (2008) 486 - 498.

[222] Shukla HS, Tamsir M, Srivastava VK, Kumar J, Approximate analytical solution of time-fractional order Cauchy-Reaction diffusion equation, CMES 103(1) (2014) 1-17.

[223] Shu C, Richards BE, Application of generalized differential quadrature to solve two dimensional incompressible navier-Stokes equations, Int. J. Numer. Methods Fluids 15 (1992) 791 798.

[224] Shu C., Generalized differential - integral quadrature and application to the simulation of incompressible viscous flows including parallel computation, PhD thesis, Univ of Glasgow, UK, (1991).

[225] Shu C, Differential Quadrature and its Application in Engineering, Athenaeum Press Ltd., Great Britain, 2000.

[226] Shu C, Chew YT, Fourier expansion - based differential quadrature and its application to Helmholtz eigenvalue problems, Commun. Numer. Methods Eng. 13 (8) (1997) 643 - 653.

[227] Shu-Sen X, Heo S, Kim S, Woo G, Yi S, Numerical solution of one dimensional Burgers' equation using reproducing kernel function, J. Comput. Appl. Math. 214 (2008) 417 - 434.

[228] Shu C, Xue H, Explicit computation of weighting coefficients in the harmonic differential quadrature, J. Sound Vib. 204 (3) (1997) 549 - 555.

[229] Siddiqui AM, Mahmood R, Ghori QK, Thin film flow of a third grade fluid on a moving belt by He's homotopy perturbation method, International Journal of Nonlinear Sciences and Numerical Simulation 7 (2006) 7-14.

[230] Soliman AA, The modified extended tanh-function method for solving Burgers-type equations, Physica A 361 (2006) 394 - 404.

[231] Shuvam S, Fourth order compact schemes for variable coefficient parabolic problems with mixed derivatives, Comput. Fluids 134-135 (2016) 81-89.

[232] Soltani LA, Shirzadi A, A new modification of the variational iteration method, Computers & Mathematics with Applications 59 (2010) 2528-2535.

[233] Siraj-ul-Islam, Haq S, Uddin M, A meshfree interpolation method for the numerical solution of the coupled nonlinear partial differential equations, Eng. Anal. Boundary Elem. 33 (2009) 399-409.

[234] Sirovich L, Turbulence and the dynamics of coherent structures. Part I. Coherent structures, Quart. Appl. Math. 45 (1987) 561-571.

[235] Sirovich L, Turbulence and the dynamics of coherent structures. Part II. Symmetries and transformations, Quart. Appl. Math. 45 (1987) 573–582.

[236] Sirovich L, Turbulence and the dynamics of coherent structures. Part III. Dynamics and scaling, Quart. Appl. Math. 45 (1987) 583–590

[237] Singh BK, Fractional reduced differential transform method for numerical computation of a system of linear and nonlinear fractional partial differential equations, Int. J. Open Problems in Comput. Sci. Math. 9(3) (2016) 20-38.

[238] Singh H, A new stable algorithm for fractional Navier-Stokes equation in Polar Coordinate, Int. J. Appl. Comput. Math (2017), http://dx.doi.org/10.1007/s40819-017-0323-7

[239] Singh BK, Arora G, A numerical scheme to solve Fisher-type reaction-diffusion equations, Nonlinear Studies/Mesa- mathematics in engineering, science and aerospace 5(2) (2014) 153-164.

[240] Singh BK, Arora G, Singh MK, A numerical scheme for the generalized Burgers-Huxley equation, J. Egypt. Math. Society (2016) http://dx.doi.org/10.1016/j.joems.2015.11.003.

[241] Singh BK, Bianca C, A new numerical approach for the solutions of partial differential equations in three-dimensional space, Appl. Math. Inf. Sci. 10(5) (2016) 1 - 10.

[242] Singh J, Kumar D, Kiliçman A, Numerical solutions of nonlinear fractional partial differential equations arising in spatial diffusion of biological populations, Abstract and Applied Analysis Volume (2014), Article ID 535793, 12 pages.

[243] Singh J, Kumar D, Swroop R, Numerical solution of time- and space-fractional coupled Burgers' equations via homotopy algorithm, Alexandria Eng. J. (2016), http://dx.doi.org/10.1016/j.aej.2016.03.028.

[244] Singh J, Kumar D, Kılıçman A, Homotopy perturbation method for fractional gas dynamics equation using sumudu transform, Abstract and Applied Analysis, Vol. 2013, Article ID 934060, 8 pages.

[245] Singh BK, Kumar P, Homotopy perturbation transform method for solving fractional partial differential equations with proportional delay, in marXiv:1611.06488, SeMA Journal (2017) https://arxiv.org/pdf/1611.06488v1.pdf

[246] Singh BK, Kumar P, Numerical computation for time-fractional gas dynamics equations by fractional reduced differential transforms method, Journal of Mathematics and System Science 6(6) (2016) 248 - 259.

[247] Singh BK, Kumar P, Extended fractional reduced differential transform for solving fractional partial differential equations with proportional delay, International Journal of Applied and Computational Mathematics (2017) doi:10.1007/s40819-017-0374-9.

[248] Singh BK, Kumar P, A novel approach for numerical computation of Burgers' equation (1 + 1) and (2 + 1) dimension, Alexandria Eng. J. (2016) http://dx.doi.org/10.1016/j.aej.2016.08.023

[249] Singh J, Kumar D, Kumar S, New treatment of fractional Fornberg - Whitham equation via Laplace transform, Ain Shams Eng J 4(3) (2013) 557 - 62.

[250] Singh J, Kumar D, Kilicman A, Application of homotopy perturbation Sumudu transform method for solving heat and wave-like equations, Malaysian J Math. Sci 7(1)(2013) 79-95.

[251] Singh BK, Mahendra, A numerical computation of a system of linear and nonlinear time dependent partial differential equations using reduced differential transform method, International Journal of Differential Equations, vol. 2016, Article ID 4275389, 8 pages, 2016.

[252] Singh BK, Srivastava VK, Approximate series solution of multi-dimensional, time fractional-order (heat-like) diffusion equations using FRDTM, R. Soc. Open Sci. 2: 140511. http://dx.doi.org/10.1098/rsos.140511.

[253] Sivashinsky GI, Instabilities, pattern-formation, and turbulence in flames, Annu Rev Fluid Mech 15 (1983) 179-99.

[254] Spiteri JR, Ruuth SJ, A new class of optimal high-order strong stability-preserving time-stepping schemes, SIAM J. Numer. Anal. 40 (2) (2002) 469 - 491.

[255] Srivastava VK, Awasthi MK, Chaurasia RK, Tamsir M, The Telegraph equation and its solution by Reduced Differential Transform Method, Modelling and Simulation in Engineering, 2013 (2013) 1-6.

[256] Srivastava VK, Mishra N, Kumar S, Singh BK, Awasthi MK, Reduced differential transform method for solving $(1+n)$-Dimensional Burgers' equation. *Egyptian Journal of Basic and Applied Sciences* **1** (2014) 115 - 119.

[257] Srivastava VK, Kumar S, Awasthi MK, Singh BK, Two-dimensional time fractional-order biological population model and its analytical solution, *Egyptian Journal of Basic and Applied Sciences* **1** (2014) 71-76.

[258] Srivastava VK, Awasthi MK, Tamsir M, RDTM solution of Caputo time fractional order hyperbolic telegraph equation, AIP Advances 3 (2013) 032142.

[259] Srivastava VK, Awasthi MK, Tamsir M, A fully implicit finite difference solution to one dimensional coupled Burgers nonlinear equation, Int. J. Math. Sci. 7 (2013) 23 - 28.

[260] Srivastava VK, Singh BK, A Robust finite difference scheme for the numerical solutions of two dimensional time - dependent coupled nonlinear Burgers' equations, International Journal of Applied Mathematics and Mechanics 10(7) (2014) 28 - 39.

[261] Srivastava VK, Singh S, Awasthi MK, Numerical solutions of coupled Burgers' equations by an implicit finite deference scheme, AIP Advances 3 (2013) 082131.

[262] Su N, Jim PCW, Keith WV, Murray EC, Mao R, Analysis of turbulent flow patterns of soil water under field conditions using Burgers equation and porous suction-cup samplers, Australian Journal of Soil Research 42(1) 9 - 16.

[263] Sushila, Singh J, Shishodia YS, A new reliable approach for two-dimensional and axisymmetric unsteady flows between parallel plates, Zeitschrift für Naturforschung A 68a (2013) 629 - 634.

[264] Sutmann G, Compact finite difference schemes of sixth order for the Helmholtz equation, J. Comput. Appl. Math. 203(1) (2007) 15-31.

[265] Swartz B, Wendroff B, The relative efficiency of finite difference and finite element methods. I: hyperbolic problems and splines SIAM J. Numer. Anal., 11(5) (1974) 979–993.

[266] Tadmor E, The well-posedness of the Kuramoto-Sivashinsky equation, SIAM J. Math. Anal. 17 (1986) 884-893.

[267] N. Taghizadeh, S.R. Moosavi Noori, Reduced differential transform method for solving parabolic-like and hyperbolic-like equations, SeMA DOI 10.1007/s40324-016-0101-1.

[268] Tamsir M, Srivastava VK, Jiwari R, An algorithm based on exponential modified cubic B-spline differential quadrature method for nonlinear Burgers' equation, App. Mathematics and Comp. 290 (2016) 111 - 124.

[269] Tari A, Shahmorad S, Differential transform method for the system of two - dimensional nonlinear Volterra integro-differential equations, Computers and Mathematics with Applications 61 (2011) 2621 - 2629.

[270] Tian Z, Liang X, Yu P, A higher order compact finite difference algorithm for solving the incompressible Navier-Stokes equations. Int. J. Numer. Methods Engg., 88(6) (2011) 511- 532.

[271] Tanthanuch J, Symmetry analysis of the nonhomogeneous inviscid burgers equation with delay, Commun. Nonlinear Sci. Numer. Simul. 17 (12) (2012) 4978 - 4987.

[272] Uddin M, Sirajul H, Siraj-ul I, A mesh-free numerical method for solution of the family of Kuramoto-Sivashinsky equations, Appl Math Comput 212 (2009) 458–69.

[273] Wang YM, Zhang HB, Higher-order compact finite difference method for systems of reaction diffusion equations, J. Comput. Appl. Math. 233 (2009) 502-518.

[274] Wirz HJ, Schutter FD, Turi A, An implicit, compact, finite difference method to solve hyperbolic equations. Math. Comput. Simulat. 19(4) (1977) 241-261.

[275] Wei GW, Zhang DS, Kouri DJ, Hoffman DK, Distributed approximation functional approach to Burgers' equation in one and two space dimensions, Comput. Phys. Commun. 111 (1998) 93 - 109.

[276] Wang K, Liu S, Analytical study of time-fractional Navier-Stokes equation by using transform methods. Advances in Difference Equations 61 (2016) http://dx.doi.org/10.1186/s13662-016-0783-9

[277] Wei GW, Gu Y, Conjugate filter approach for solving Burgers' equation, J. Comput. Appl. Math. 149(2) (2002) 439 - 56.

[278] Weinan E, Liu JG, Essentially compact schemes for unsteady viscous incompressible flows, J. Comput. Phys. 126 (1996) 122-138.

[279] Wu J, Theory and Applications of Partial Functional Differential Equations, Springer–Verlag, New York, 1996 .

[280] Wu GC, Lee EWM, Fractional variational iteration method and its application, Physics Letters A (2010) doi:10.1016/j.physleta.2010.04.034.

[281] Xu L, He's homotopy perturbation method for a boundary layer equation in unbounded domain, Computers & Mathematics with Applications 54 (2007) 1067–1070.

[282] Xu M, Wang R, Zhang J, Fang Q, A novel numerical scheme for solving Burgers' equation, Appl. Math. Comput. 217(2011) 4473-4482.

[283] Yang XJ, Baleanu D, Khan Y, Mohyud-din ST, Local fractional variational iteration method for diffusion and wave equations on cantor sets, Rom. J. Phys. 59 (1-2) (2014) 36-48.

[284] Yan, X, Shu, CW, Local discontinuous Galerkin methods for the Kuramoto-Sivashinsky equations and the Ito-type coupled KdV equations, Comput. Methods Appl. Mech. Eng. 195 (2006) 3430-3447.

[285] Ye, L, Yan, G, Li, T, Numerical method based on the Lattice Boltzmann model for the Kuramoto-Sivashinsky equation, J Sci Comput (2011), http://dx.doi.org/10.1007/s10915-010-9455-1.

[286] Yıldırım A, Kelleci A, Homotopy perturbation method for numerical solutions of coupled Burgers equations with time -and space fractional derivatives, Int. J. Numer. Methods Heat & Fluid Flow 20(8) (2010) 897–909.

[287] Yang XJ, Tenreiro Machado JA, Baleanu D, Carlo C, On exact traveling-wave solutions for local fractional Korteweg-de Vries equation, Chaos: An Interdisciplinary Journal of Nonlinear Science 26(8) 084312.

[288] Yang XJ, Tenreiro Machado JA, Srivastava HM, A new numerical technique for solving the local fractional diffusion equation: Two-dimensional extended differential transform approach, Appl. Math. Compu. 274 (2016) 143-151.

[289] Yang XJ, Srivastava HM, An asymptotic perturbation solution for a linear oscillator of free damped vibrations in fractal medium described by local fractional derivatives, Commu. Nonlinear Sci. Numer. Simul. 29 (2015) 499–504.

[290] Yang XJ, Srivastava HM, Tenreiro Machado JA, A new fractional derivative without singular kernel: Application to the modelling of the steady heat flow, Thermal Science, 20(2) (2016) 753-756.

[291] Yong C, Hong-Li A, Numerical solutions of coupled Burgers equations with time and space-fractional derivatives, Applied Mathematics and Computation 200 (2008) 87-95.

[292] Yu J, Jing J, Sun Y, Wu S, $(n+1)$ dimensional reduced differential transform method for solving partial differential equations, Applied Math. Comput. 273 (2016) 697 - 705.

[293] Yusufoglu E, Numerical solution of Duffing equation by the Laplace decomposition algorithm, Applied Mathematics and Computation 177 (2006) 572–580.

[294] Yasir Khan, An effective modification of the Laplace decomposition method for nonlinear equations, International Journal of Nonlinear Sciences and Numerical Simulation 10 (2009) 1373–1376.

[295] Zarebnia M, Parvaz, R, Septic B-spline collocation method for numerical solution of the Kuramoto-Sivashinsky equation, Int. J. Math. Comput. Sci. Engg. 2(1) (2013) 55-61.

[296] Zhao J, Corless RM, Compact finite difference method for integro-differential equations, Appl. Math. Comput. 177(1) ( 2006) 271-288.

[297] Zhang L, Ouyang J, Wang X, Zhang X, Variational multiscale element-free Galerkin method for $2D$ Burgers equation, J. Comput. Phy. 229(19) (2010) 7147-7161.

[298] Zhang RP, Jun XJ, Zhao GZ, Local discontinuous Galerkin method for solving Burgers and coupled Burgers equations, Chin. Phys. B 20 (2011) 110205.

[299] Zubik-Kowal B, Chebyshev pseudospectral method and waveform relaxation for differential and differential-functional parabolic equations, Appl. Numer. Math. 34 (2-3) (2000) 309-328.

[300] Zubik-Kowal B, Jackiewicz Z, Spectral collocation and waveform relaxation methods for nonlinear delay partial differential equations, Appl. Numer. Math. 56 (3-4) (2006) 433-443.

[301] Zhou JK, Differential Transformation and its Application for Electrical Circuits, Huazhong University Press, Wuhan, China, 1986 (in Chinese).

Ain Shams Engineering Journal (2016) xxx, xxx–xxx

Ain Shams University

Ain Shams Engineering Journal

www.elsevier.com/locate/asej
www.sciencedirect.com

ENGINEERING PHYSICS AND MATHEMATICS

# FRDTM for numerical simulation of multi-dimensional, time-fractional model of Navier–Stokes equation

**Brajesh Kumar Singh** *, **Pramod Kumar**

*Department of Applied Mathematics, School for Physical Sciences, Babasaheb Bhimrao Ambedkar University, Lucknow 226 025, UP, India*

Received 27 September 2015; revised 2 April 2016; accepted 29 April 2016

**KEYWORDS**

Navier–Stokes equation;
Caputo time-fractional derivative;
FRDTM;
Mittag–Leffler function

**Abstract** In this paper, a new approximate solution of time-fractional order multi-dimensional Navier–Stokes equation is obtained by adopting a semi-analytical scheme: "Fractional Reduced Differential Transformation Method (FRDTM)". Three test problems are carried out in order to validate and illustrate the efficiency of the method. The scheme is found to be very reliable, effective and efficient powerful technique to solve wide range of problems arising in engineering and sciences. The small size of computation contrary to the other schemes, is its strength.



## 1. Introduction

The idea of fractional derivative was first given by a great mathematician Leibniz, in 1695, in a letter to L'Hospital. Fractional calculus deals with the differential and integral operators with non-integral powers. Noting that the integer-order differential operator is a local operator while the fractional-order differential operator is non-local, it means that the next state of a system depends not only upon its current state but also upon all of its previous states. It is more realistic and is one of the main reasons why the fractional calculus has become so popular. In the recent years, advances of fractional differential equations have a great attention due to their numerous applications in a wide range of nonlinear complex systems arising in fluid mechanics, viscoelasticity, mathematical biology, life sciences, electrochemistry and physics [1–8]. For instance, the non-linear oscillation of earthquake can be modeled with fractional derivatives [9], and the fluid-dynamic traffic model with fractional derivatives [10] can eliminate the deficiency arising from the assumption of continuum traffic flow. Based on experimental data fractional partial differential equations for seepage flow in porous media are suggested in [11]. Fractional differential equations have created attention among the researcher due to exact description of non-linear phenomena, especially in nano-hydrodynamics where continuum assumption does not well, and fractional model can be considered to be a best candidate. These findings

* Corresponding author.
E-mail addresses: bksingh0584@gmail.com (B.K. Singh), bbaupramod@gmail.com (P. Kumar).

Peer review under responsibility of Ain Shams University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.asej.2016.04.009



Please cite this article in press as: Singh BK, Kumar P, FRDTM for numerical simulation of multi-dimensional, time-fractional model of Navier–Stokes equation, Ain Shams Eng J (2016), http://dx.doi.org/10.1016/j.asej.2016.04.009

invoked the growing interest of studies of the fractal calculus in many branches of science and engineering.

In the recent various analytical techniques such as Homotopy perturbation method (HPM) [10], homotopy perturbation Sumudu transform method [12,13], homotopy analysis method (HAM) [14,15] and Adomian decomposition method (ADM) [16,17] have been developed to solve the fractional partial differential equations. By coupling of HPM and Laplace transform algorithm (LTA), Kumar et al. solved analytically the nonlinear fractional Zakharov–Kuznetsov equation in [18]. At first, Keskin and Oturanc [19] introduce reduced differential transform method (RDTM) as a reduced form of differential transform method, and implement it to find the approximate solutions of partial (and factional partial) differential equations [19,20]. Fractional reduced differential transform method (FRDTM) has been adopted in many articles to solve the differential equations prevailing in mathematics, physics and engineering [21–36].

A famous governing equation of motion of viscus fluid flow called Navier–Stokes (NS) equation has been derived in 1822 [37]. The equation can be regarded as Newton's second law of motion for fluid substances, and is a combination of Momentum equation, continuity equation and the energy equation. This equation describes many physical things such as ocean currents, liquid flow in pipes, blood flow and air flow around the wings of an aircraft. The fractional modeling of NS equations was first done in 2005 by El-Shahed and Salem [38]. The authors [38] generalized the classical NS equations using Laplace transform, finite Hankel transforms and finite Fourier Sine transform. By coupling of HPM and LTA, Kumar et al. [39] solved analytically a nonlinear fractional model of NS equation. Ragab et al. [14] and Ganji et al. [15] solved nonlinear time-fractional NS equation by adopting HAM. Birajdar [16] and Momani and Odibat [17] adopted ADM for numerical computation of time-fractional NS equation. Analytical solution of time-fractional NS equation is obtained using coupling of ADM and LTA by Kumar et al.[40] while Chaurasia and Kumar [41] solved the same equation by coupling of Laplace transform and finite Hankel transform. This paper presents an approximate analytic solution of multi-dimensional, time-fractional model of NS equation by adopting FRDTM.

The rest of the paper is organized as follows: some basic definitions and notations on fractional calculus are revisited in Section 2 while the preliminary on FRDTM is presented in Section 2.1. In Section 3.1, the approximate analytic solutions of three test problems of time-fractional order NS equation are obtained. Section 4 concludes the study.

## 2. Fractional calculus theory: basic definitions and notations

In this section, among several definitions of fractional integrals or fractional derivatives, available in the literature due to Riemann-Liouville, Grunwald-Letnikov, Caputo, etc., only those basic definitions and preliminaries are revisited, which we need to complete our study.

**Definition 1** ([1,2]). Let $\mu \in \mathbb{R}$ and $m \in \mathbb{N}$. A real valued function $f: \mathbb{R}^+ \to \mathbb{R}$ belongs to $\mathbb{C}_\mu$ if there exists $k \in \mathbb{R}$, $k > \mu$ and $g \in C[0,\infty)$ such that $f(x) = x^k g(x)$, for all $x \in \mathbb{R}^+$. Moreover, $f \in \mathbb{C}_\mu^m$ if $f^{(m)} \in \mathbb{C}_\mu$.

**Definition 2** ([1,2]). The Riemann–Liouville fractional integral of $f \in \mathbb{C}_\mu$ of the order $\alpha \geqslant 0$ is defined as

$$J_t^\alpha f(t) = \begin{cases} f(t) & \text{if } \alpha = 0, \\ \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau) d\tau, & \text{if } \alpha > 0, \end{cases} \tag{1}$$

where $\Gamma$ *denotes* gamma function: $\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt, z \in \mathbb{C}$.

In their work, Caputo and Mainardi [3] proposed a modified fractional differentiation operator $D_t^\alpha$ to describe the theory of viscoelasticity in order to overcome the discrepancy of Riemann–Liouville derivative [1,2]. It is mentioned that the proposed Caputo fractional derivative allows the utilization of initial and boundary conditions involving integer order derivatives.

**Definition 3** ([1,3]). The fractional derivative of $f \in \mathbb{C}_\mu$ of the order $\alpha \geqslant 0$, in Caputo sense, is defined as

$$\begin{aligned} D_t^\alpha f(t) &= J_t^{m-\alpha} D_t^m f(t) \\ &= \frac{1}{\Gamma(m-\alpha)} \int_0^t (t-\tau)^{m-\alpha-1} f^{(m)}(\tau) d\tau, \end{aligned} \tag{2}$$

for $m-1 < \alpha \leqslant m$, $m \in \mathbb{N}$, $t > 0$, $f \in \mathbb{C}_\mu^m$, $\mu \geqslant -1$.

The basic properties of Caputo fractional derivative are given as follows:

**Lemma 1** ([1–4]). *Let* $m-1 < \alpha \leqslant m, m \in \mathbb{N}$, *and* $f \in \mathbb{C}_\mu^m$, $\mu \geqslant -1$, *then*

$$D_t^\alpha J_t^\alpha f(t) = f(t)$$

$$J_t^\alpha D_t^\alpha f(t) = f(t) - \sum_{k=0}^{m} f^{(k)}(0^+) \frac{t^k}{k!}, \quad \text{for } t > 0.$$

In the present work, Caputo fractional derivative is considered because it includes traditional initial and boundary conditions in the formulation of the physical problems. For more details on fractional derivatives, one can refer [1–5].

### 2.1. Fractional reduced differential transform method (FRDTM)

This section describes the basic properties of fractional reduced differential transform method [25,26]. Let $\psi(x,t)$ be a function of two variables such that $\psi(x,t) = f(x)g(t)$, then from the properties of one-dimensional differential transform (DT) method, we have

$$\psi(x,t) = \sum_{i=0}^{\infty} f(i) x^i \sum_{j=0}^{\infty} g(j) t^j = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \Psi(i,j) x^i t^j, \tag{3}$$

where $\psi(i,j) = f(i)g(j)$ is referred as the spectrum of $\psi(x,t)$. Throughout the paper $R_D$ and $R_D^{-1}$ denote the operators for fractional reduced differential transform (FRDT) and inverse FRDT, respectively. Further, the lowercase $\psi(x,t)$ is used for the original function whereas its fractional reduced transformed function is represented by the uppercase $\Psi_k(x)$.

The basic definitions and properties of FRDTM are described below.

Please cite this article in press as: Singh BK, Kumar P, FRDTM for numerical simulation of multi-dimensional, time-fractional model of Navier–Stokes equation, Ain Shams Eng J (2016), http://dx.doi.org/10.1016/j.asej.2016.04.009

**Definition 4** ([25,26]). Let $\psi(x,t)$ be an analytic and continuously differentiable with respect to space variable $x$ and time variable $t$ in the domain of interest, then

(a) FRDT of $\psi$ is given by

$$\Psi_k(x) = \frac{1}{\Gamma(k\alpha+1)}\left[D_t^{\alpha k}(\psi(x,t))\right]_{t=t_0}, \quad k = 0,1,2,\ldots$$

where $\alpha$ describes the order of time-fractional derivative.

(b) The inverse FRDT of $\Psi_k(x)$ is defined by

$$\psi(x,t) = \sum_{k=0}^{\infty}\Psi_k(x)(t-t_0)^{k\alpha}.$$

(c) From (a) and (b), we have

$$\psi(x,t) = \sum_{k=0}^{\infty}\frac{1}{\Gamma(k\alpha+1)}\left[D_t^{\alpha k}(\psi(x,t))\right]_{t=t_0}(t-t_0)^{k\alpha}.$$

In particular, for $t_0 = 0$, above equation becomes

$$\psi(x,t) = \sum_{k=0}^{\infty}\frac{1}{\Gamma(k\alpha+1)}\left[D_t^{\alpha k}(\psi(x,t))\right]_{t=0}t^{k\alpha}.$$

It shows that FRDTM is a generalization of the power series expansion.

**Theorem 1** ([24–26]). *Let $u(x,t)$ and $v(x,t)$ be any two analytic and continuously differentiable functions with respect to space variable $x$ and time $t$ such that $u(x,t) = R_D^{-1}[U_k(x)]$ and $v(x,t) = R_D^{-1}[V_k(x)]$, then*

(a) $R_D\{u(x,t)v(x,t)\} = U_k(x) \otimes V_k(x) = \sum_{r=0}^{k} U_r(x)V_{k-r}(x)$;
(b) $R_D\{a_1u(x,t) \pm a_2v(x,t)\} = a_1U_k(x) \pm a_2V_k(x)$;
(c) $R_D\{x^m t^n u(x,t)\} = \begin{cases} x^m U_{k-n}(x) & \text{if } k \geqslant n \\ 0, & \text{else} \end{cases}$;
(d) $R_D\{D_t^{N\alpha}(u(x,t))\} = \frac{\Gamma(1+(k+N)\alpha)}{\Gamma(1+k\alpha)}U_{k+N}(x)$;
(e) $R_D\{D_x^l u(x,t)\} = D_x^l U_k(x)$; $R_D\{x^m\} = x^m\delta(k)$; & $R_D\{e^{\lambda t}\} = \frac{\lambda^k}{k!}$,

*where the convolution $\otimes$ denotes the fractional reduced differential transform version of multiplication and the function $\delta$ is defined by* $\delta(k) = \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{otherwise} \end{cases}$.

## 3. Implementation of FRDTM on Navier–Stokes equation

In this section, the numerical study of time-fractional model of NS equation of order $\alpha(\alpha \leqslant 1)$ is presented. The time-fractional model of NS equation for an incompressible fluid flow of kinematic viscosity $v = \eta/\rho$ and constant density $\rho$ is given as follows [16,37]:

$$\begin{cases} D_t^{\alpha}U + (U\cdot\nabla)U = \rho_0\nabla^2 U - \frac{1}{\rho}\nabla p, & \text{on } \Omega\times(0,T) \\ \nabla\cdot U = 0, & \text{on } \Omega\times(0,T) \\ U = 0, & \text{on } \partial\Omega\times(0,T) \end{cases} \tag{4}$$

where $U = (u,v,w)$, $t$, $p$ denote the fluid vector, time and the pressure, respectively. $(x,y,z)$ are spatial components in $\Omega$ and $\partial\Omega$ is the boundary of $\Omega$, $\eta$ denotes dynamic viscosity and $\rho$ is the density while the ratio $\rho_0 = \eta/\rho$ denotes the kinematic viscosity of the flow. In Cartesian co-ordinates, the above equation becomes

$$\begin{cases} D_t^{\alpha}u + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} = \rho_0\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) - \frac{1}{\rho}\frac{\partial p}{\partial x}, \\ D_t^{\alpha}v + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} = \rho_0\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right) - \frac{1}{\rho}\frac{\partial p}{\partial y}, \\ D_t^{\alpha}w + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} = \rho_0\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) - \frac{1}{\rho}\frac{\partial p}{\partial z}, \end{cases} \tag{5}$$

Further, if $p$ is known, then $g_1 = -\frac{1}{\rho}\frac{\partial p}{\partial x}$, $g_2 = -\frac{1}{\rho}\frac{\partial p}{\partial y}$, $g_3 = -\frac{1}{\rho}\frac{\partial p}{\partial z}$ can be determined. Applying FRDTM on Eq. (5), we have

$$\begin{cases} \frac{\Gamma(1+(1+k)\alpha)}{\Gamma(1+k\alpha)}U_{k+1} + \sum_{\ell=0}^{k}\left(\frac{\partial U_\ell}{\partial x}U_{k-\ell} + \frac{\partial U_\ell}{\partial y}V_{k-\ell} + \frac{\partial U_\ell}{\partial z}W_{k-\ell}\right) \\ \qquad = \rho_0\nabla^2(U_k) + g_1\delta(k), \\ \frac{\Gamma(1+(1+k)\alpha)}{\Gamma(1+k\alpha)}V_{k+1} + \sum_{\ell=0}^{k}\left(\frac{\partial V_\ell}{\partial x}U_{k-\ell} + \frac{\partial V_\ell}{\partial y}V_{k-\ell} + \frac{\partial V_\ell}{\partial z}W_{k-\ell}\right) \\ \qquad = \rho_0\nabla^2(V_k) + g_2\delta(k), \\ \frac{\Gamma(1+(1+k)\alpha)}{\Gamma(1+k\alpha)}W_{k+1} + \sum_{\ell=0}^{k}\left(\frac{\partial W_\ell}{\partial x}U_{k-\ell} + \frac{\partial W_\ell}{\partial y}V_{k-\ell} + \frac{\partial W_\ell}{\partial z}W_{k-\ell}\right) \\ \qquad = \rho_0\nabla^2(W_k) + g_3\delta(k), \end{cases} \tag{6}$$

where $\nabla^2 \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$, and $U_k = U_k(x,y,z)$, etc. One can obtain the recursive values of $U_k$, $V_k$, $W_k$ by solving above equation simultaneously once the values $U_0$, $V_0$, $W_0$ are known.

### 3.1. Illustrative examples

**Example 1.** Consider time-fractional order 2-dimensional NS equation with $g_1 = -g_2 = g$ as

$$\begin{aligned} D_t^{\alpha}u + u\frac{\partial u}{\partial x} + u\frac{\partial v}{\partial y} &= \rho_0\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + g, \\ D_t^{\alpha}v + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} &= \rho_0\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) - g, \end{aligned} \tag{7}$$

subject to the initial condition

$$u(x,y,0) = -\sin(x+y), \quad v(x,y,0) = \sin(x+y), \tag{8}$$

Using FRDTM on the above two equations, we obtained the following recurrence relation:

$$\begin{cases} \frac{\Gamma(1+(1+k)\alpha)}{\Gamma(1+k\alpha)}U_{k+1} + \sum_{\ell=0}^{k}\left(\frac{\partial U_\ell}{\partial x}U_{k-\ell} + \frac{\partial U_\ell}{\partial y}V_{k-\ell}\right) = \rho_0\nabla^2(U_k) + g\delta(k), \\ \frac{\Gamma(1+(1+k)\alpha)}{\Gamma(1+k\alpha)}V_{k+1} + \sum_{\ell=0}^{k}\left(\frac{\partial V_\ell}{\partial x}U_{k-\ell} + \frac{\partial V_\ell}{\partial y}V_{k-\ell}\right) = \rho_0\nabla^2(V_k) - g\delta(k), \\ U_0 = -\sin(x+y), \quad V_0 = \sin(x+y) \end{cases} \tag{9}$$

On solving the system (9), we have

$$
\begin{aligned}
&U_1(x,y) = \tfrac{2\rho_0}{\Gamma(1+\alpha)}\sin(x+y) + \tfrac{g}{\Gamma(1+\alpha)};\\
&V_1(x,y) = -\tfrac{2\rho_0}{\Gamma(1+\alpha)}\sin(x+y) - \tfrac{g}{\Gamma(1+\alpha)}\\
&U_2(x,y) = -\tfrac{(2\rho_0)^2}{\Gamma(1+2\alpha)}\sin(x+y); \quad V_2(x,y) = \tfrac{(2\rho_0)^2}{\Gamma(1+2\alpha)}\sin(x+y)\\
&U_3(x,y) = \tfrac{(2\rho_0)^3}{\Gamma(1+3\alpha)}\sin(x+y); \quad V_3(x,y) = -\tfrac{(2\rho_0)^3}{\Gamma(1+3\alpha)}\sin(x+y)\\
&\vdots \qquad \vdots \quad \vdots \qquad \vdots\\
&U_k(x,y) = -\tfrac{(-2\rho_0)^k}{\Gamma(1+k\alpha)}\sin(x+y); \quad V_k(x,y) = \tfrac{(-2\rho_0)^k}{\Gamma(1+k\alpha)}\sin(x+y),\\
&\text{for any integer } k \geqslant 2,
\end{aligned}
\tag{10}
$$

By using inverse FRDT, we have

$$
\begin{aligned}
u(x,y,t) &= \sum_{k=0}^{\infty} U_k(x,y)t^{\alpha k} = -\sin(x+y)\sum_{k=0}^{\infty}\tfrac{(-2\rho_0 t^\alpha)^k}{\Gamma(1+k\alpha)} + \tfrac{gt^\alpha}{\Gamma(1+\alpha)},\\
&= -\sin(x+y)E_{\alpha,1}(-2\rho_0 t^\alpha) + \tfrac{gt^\alpha}{\Gamma(1+\alpha)},
\end{aligned}
\tag{11}
$$

$$
\begin{aligned}
v(x,y,t) &= \sum_{k=0}^{\infty} V_k(x,y)t^{\alpha k} = \sin(x+y)\sum_{k=0}^{\infty}\tfrac{(-2\rho_0 t^\alpha)^k}{\Gamma(1+k\alpha)} - \tfrac{gt^\alpha}{\Gamma(1+\alpha)},\\
&= \sin(x+y)E_{\alpha,1}(-2\rho_0 t^\alpha) - \tfrac{gt^\alpha}{\Gamma(1+\alpha)}.
\end{aligned}
\tag{12}
$$

where $E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty}\frac{z^k}{\Gamma(\beta+k\alpha)}$, for $\alpha, \beta > 0$ denotes the Mittag–Leffler function with two parameters [1], and notice that $E_{1,1}(z) = e^z$. For $g = 0$ and $\alpha = 1$ Eqs. (11), (12) reduces to

$$
\begin{aligned}
&u(x,y,t) = -e^{-2\rho_0 t}\sin(x+y);\\
&v(x,y,t) = e^{-2\rho_0 t}\sin(x+y),
\end{aligned}
\tag{13}
$$

which is the exact solution of classical NS equation for the velocity field. The behavior of velocity field of the classical NS equation is depicted in Fig. 1, and the behavior of NS equation with time-fraction order $\alpha = 0.1$, 0.5 and 0.8 is depicted in Figs. 2–4, respectively.

**Example 2.** Consider time-fractional order two dimensional NS Eq. (7) subject to the initial condition

$$
u(x,y,0) = -e^{x+y}, \quad v(x,y,0) = e^{x+y},
\tag{14}
$$

Using FRDTM on Eqs. (7) and (14), we obtained the following recurrence relation:

$$
\begin{cases}
\frac{\Gamma(1+(1+k)\alpha)}{\Gamma(1+k\alpha)}U_{k+1} + \sum_{\ell=0}^{k}\left(\frac{\partial U_\ell}{\partial x}U_{k-\ell} + \frac{\partial U_\ell}{\partial y}V_{k-\ell}\right) = \rho_0\nabla^2(U_k) + g_1\delta(k),\\
\frac{\Gamma(1+(1+k)\alpha)}{\Gamma(1+k\alpha)}V_{k+1} + \sum_{\ell=0}^{k}\left(\frac{\partial V_\ell}{\partial x}U_{k-\ell} + \frac{\partial V_\ell}{\partial y}V_{k-\ell}\right) = \rho_0\nabla^2(V_k) - g_1\delta(k),\\
U_0 = -e^{x+y}, \quad V_0 = e^{x+y}.
\end{cases}
\tag{15}
$$

On solving the system (15), we have

$$
\begin{aligned}
&U_1(x,y) = -\tfrac{2\rho_0}{\Gamma(1+\alpha)}e^{x+y} + \tfrac{g}{\Gamma(1+\alpha)}; \quad V_1(x,y) = \tfrac{2\rho_0}{\Gamma(1+\alpha)}e^{x+y} - \tfrac{g}{\Gamma(1+\alpha)}\\
&U_2(x,y) = -\tfrac{(2\rho_0)^2}{\Gamma(1+2\alpha)}e^{x+y}; \quad V_2(x,y) = \tfrac{(2\rho_0)^2}{\Gamma(1+2\alpha)}e^{x+y}\\
&U_3(x,y) = -\tfrac{(2\rho_0)^3}{\Gamma(1+3\alpha)}e^{x+y}; \quad V_3(x,y) = \tfrac{(2\rho_0)^3}{\Gamma(1+3\alpha)}e^{x+y}\\
&\vdots \qquad \vdots \quad \vdots \qquad \vdots\\
&U_k(x,y) = -\tfrac{(2\rho_0)^k}{\Gamma(1+k\alpha)}e^{x+y}; \quad V_k(x,y) = \tfrac{(2\rho_0)^k}{\Gamma(1+k\alpha)}e^{x+y}, \ \forall k \geqslant 2
\end{aligned}
\tag{16}
$$

By using inverse FRDT, we have

$$
\begin{aligned}
u(x,y,t) &= \sum_{k=0}^{\infty} U_k(x,y)t^{\alpha k} = -e^{x+y}\sum_{k=0}^{\infty}\tfrac{(2\rho_0 t^\alpha)^k}{\Gamma(1+k\alpha)} + \tfrac{gt^\alpha}{\Gamma(1+\alpha)},\\
&= -e^{x+y}E_{\alpha,1}(2\rho_0 t^\alpha) + \tfrac{gt^\alpha}{\Gamma(1+\alpha)},
\end{aligned}
\tag{17}
$$

$$
\begin{aligned}
v(x,y,t) &= \sum_{k=0}^{\infty} V_k(x,y)t^{\alpha k} = e^{x+y}\sum_{k=0}^{\infty}\tfrac{(2\rho_0 t^\alpha)^k}{\Gamma(1+k\alpha)} - \tfrac{gt^\alpha}{\Gamma(1+\alpha)},\\
&= e^{x+y}E_{\alpha,1}(2\rho_0 t^\alpha) - \tfrac{gt^\alpha}{\Gamma(1+\alpha)}.
\end{aligned}
\tag{18}
$$

With $g = 0$, the solution of the problem is obtained by Birajdar [16]. For $g = 0$ and $\alpha = 1$, we have

$$
u(x,y,t) = -e^{x+y+2\rho_0 t}, \quad v(x,y,t) = e^{x+y+2\rho_0 t},
\tag{19}
$$

which is the exact solution of classical NS equation for the velocity field. The behavior of velocity field of the NS equation is depicted for $\alpha = 1$ and 0.5 in Figs. 5 and 6, respectively.

**Example 3.** Consider time-fractional order three dimensional NS Eq. (5) with $g_1 = g_2 = g_3 = 0$ subject to the initial condition

$$
\begin{aligned}
&u(x,y,z,0) = -0.5x + y + z, \quad v(x,y,z,0) = x - 0.5y + z,\\
&w(x,y,z,0) = x + y - 0.5z.
\end{aligned}
\tag{20}
$$

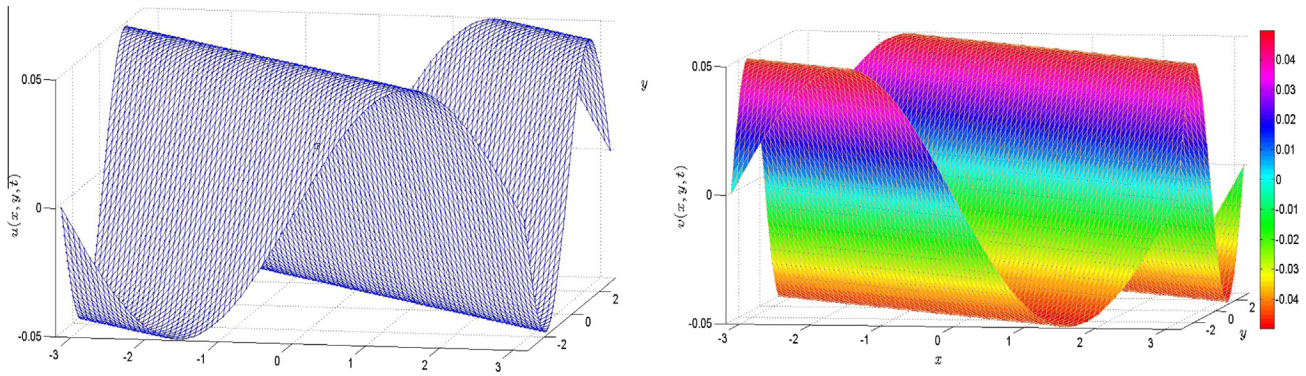


**Figure 1** The behavior of $u$ and $v$ of NS equation in Example 1 at $t = 3$ with the parameters $\alpha = 1$, $g = 0$, $\rho_0 = 0.5$.

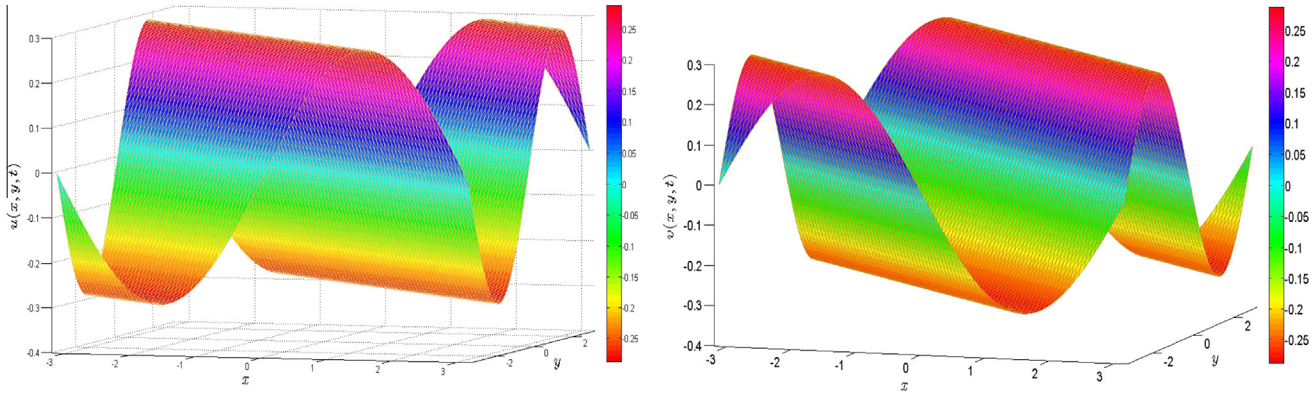

**Figure 2** The behavior of $u$ and $v$ of NS equation in Example 1 at $t = 3$ with the parameters $\alpha = 0.5,\ g = 0,\ \rho_0 = 0.5$.

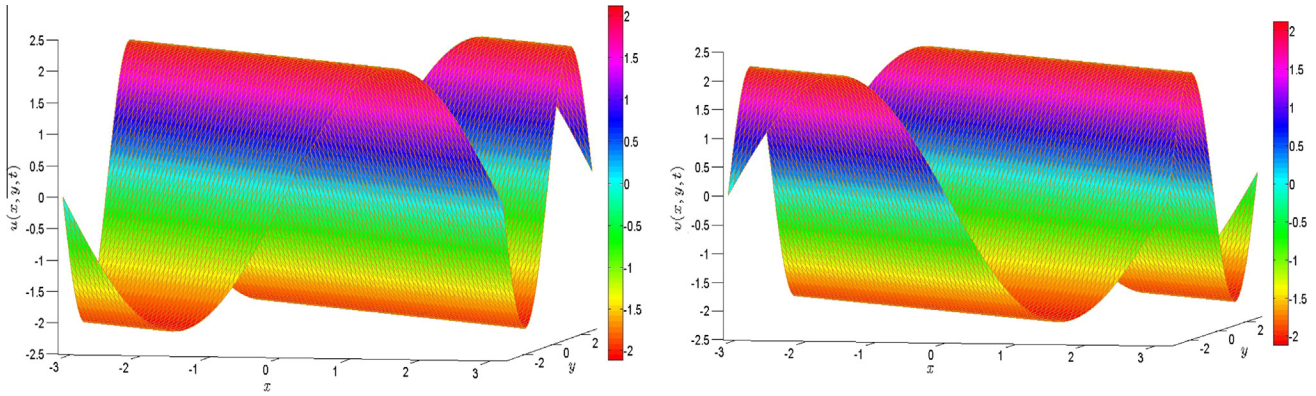

**Figure 3** The behavior of $u$ and $v$ of NS equation in Example 1 at $t = 3$ with the parameters $\alpha = 0.1,\ g = 0,\ \rho_0 = 0.5$.

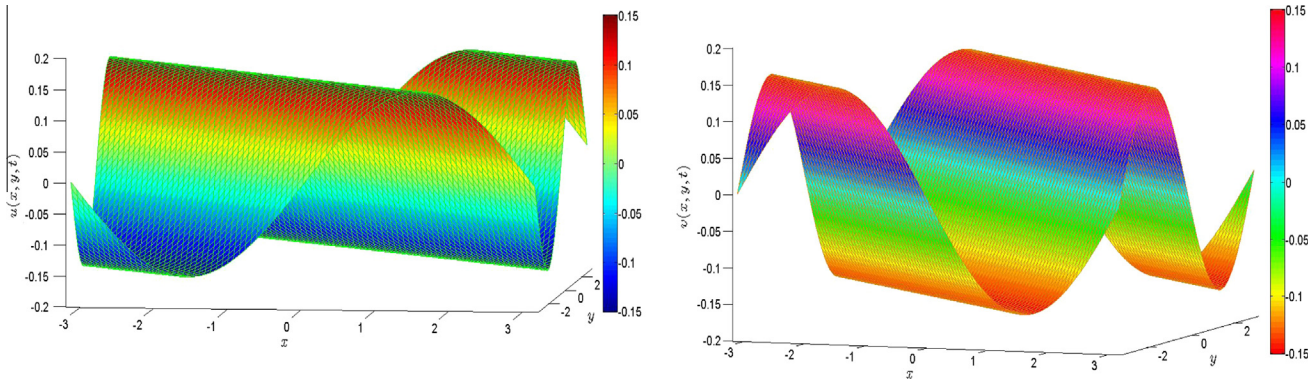

**Figure 4** The behavior of $u$ and $v$ of NS equation in Example 1 at $t = 3$ with the parameters $\alpha = 0.8,\ g = 0,\ \rho_0 = 0.5$.

Using FRDTM on these equations, we obtained the following recurrence relation:

$$\begin{cases} \frac{\Gamma(1+(1+k)\alpha)}{\Gamma(1+k\alpha)} U_{k+1} + \sum_{\ell=0}^{k} \left( \frac{\partial U_\ell}{\partial x} U_{k-\ell} + \frac{\partial U_\ell}{\partial y} V_{k-\ell} + \frac{\partial U_\ell}{\partial z} W_{k-\ell} \right) = \rho_0 \nabla^2 (U_k), \\ \frac{\Gamma(1+(1+k)\alpha)}{\Gamma(1+k\alpha)} V_{k+1} + \sum_{\ell=0}^{k} \left( \frac{\partial V_\ell}{\partial x} U_{k-\ell} + \frac{\partial V_\ell}{\partial y} V_{k-\ell} + \frac{\partial V_\ell}{\partial z} W_{k-\ell} \right) = \rho_0 \nabla^2 (V_k), \\ \frac{\Gamma(1+(1+k)\alpha)}{\Gamma(1+k\alpha)} W_{k+1} + \sum_{\ell=0}^{k} \left( \frac{\partial W_\ell}{\partial x} U_{k-\ell} + \frac{\partial W_\ell}{\partial y} V_{k-\ell} + \frac{\partial W_\ell}{\partial z} W_{k-\ell} \right) = \rho_0 \nabla^2 (W_k), \\ U_0(x,y,z) = -0.5x + y + z, \quad V_0(x,y,z) = x - 0.5y + z, \quad W_0(x,y,z) = x + y - 0.5z. \end{cases} \tag{21}$$

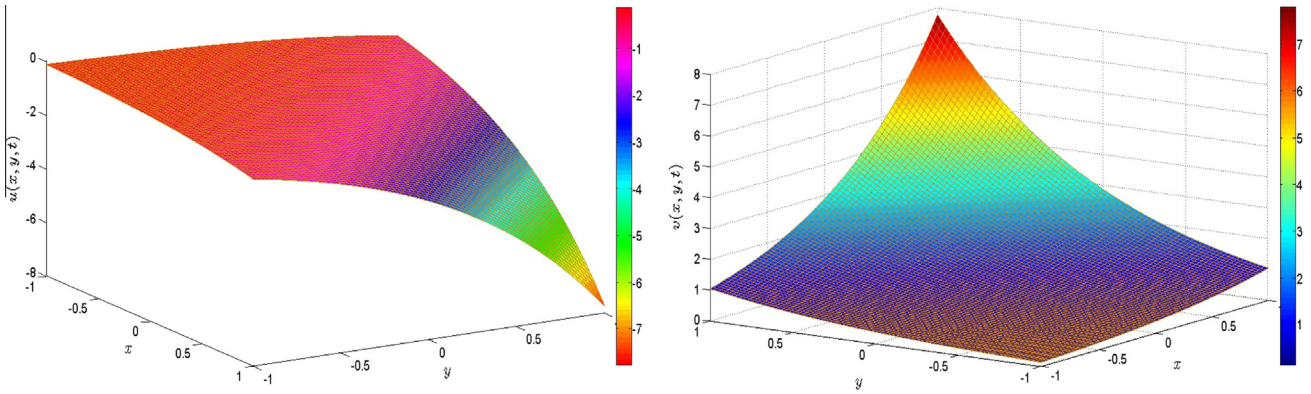

**Figure 5** The behavior of $u$ and $v$ of NS equation in Example 2 with the parameters $\alpha = 1,\ g = 0,\ \rho_0 = 0.5$ at $t = 0.05$.

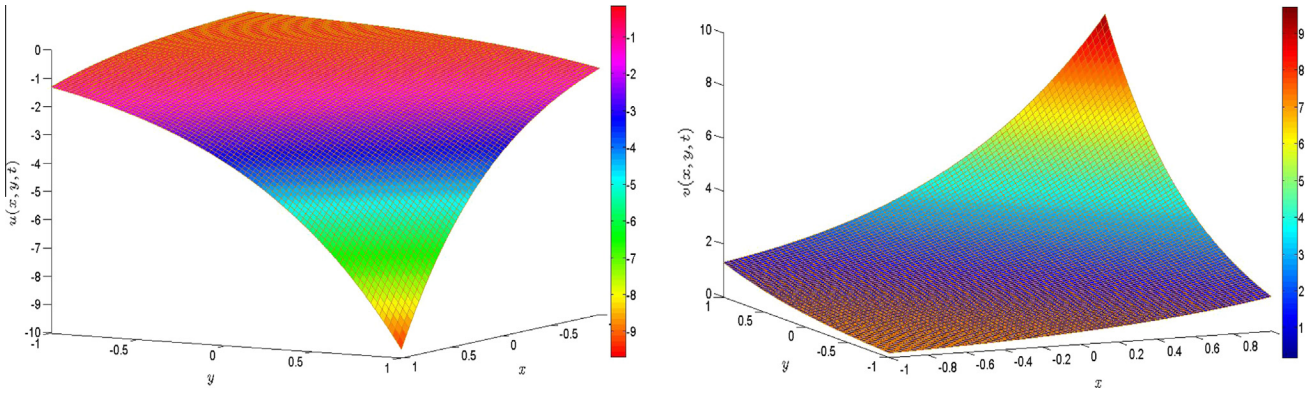

**Figure 6** The behavior of $u$ and $v$ of NS equation in Example 2 with the parameters $\alpha = 0.5,\ g = 0,\ \rho_0 = 0.5$ at $t = 0.05$.

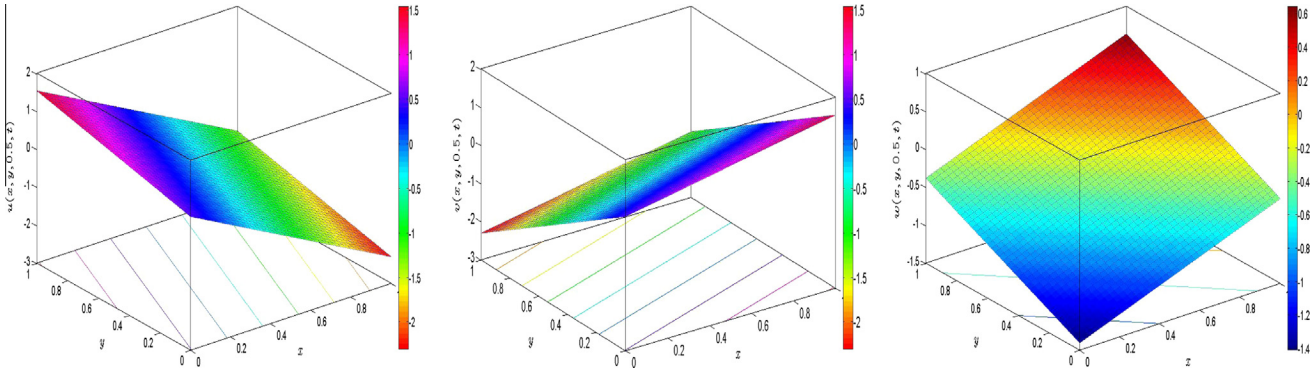

**Figure 7** The velocity profile $(u, v, w)$ of NS equation in Example 3 at $t = 0.1$ with $\alpha = 1$.

On solving the simultaneous equations in (21) and denoting $U_k(x, y, z) \equiv U_k$, etc., we have

$$
\begin{aligned}
&U_1 = -\tfrac{2.25}{\Gamma(1+\alpha)}x; \quad U_2 = \tfrac{2(2.25)}{\Gamma(1+2\alpha)}U_0; \quad U_3 = -\tfrac{(2.25)^2}{\Gamma(1+3\alpha)}\left(4 + \tfrac{\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2}\right)x; U_4 = \tfrac{(2.25)^2}{\Gamma(1+4\alpha)}\left(8 + \tfrac{2\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2} + \tfrac{4\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)}\right)U_0; \ldots \\
&V_1 = -\tfrac{2.25}{\Gamma(1+\alpha)}y; \quad V_2 = \tfrac{2(2.25)}{\Gamma(1+2\alpha)}V_0; \quad V_3 = -\tfrac{(2.25)^2}{\Gamma(1+3\alpha)}\left(4 + \tfrac{\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2}\right)y; V_4 = \tfrac{(2.25)^2}{\Gamma(1+4\alpha)}\left(8 + \tfrac{2\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2} + \tfrac{4\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)}\right)V_0; \ldots \\
&W_1 = -\tfrac{2.25}{\Gamma(1+\alpha)}z; \quad W_2 = \tfrac{2(2.25)}{\Gamma(1+2\alpha)}W_0; \quad W_3 = -\tfrac{(2.25)^2}{\Gamma(1+3\alpha)}\left(4 + \tfrac{\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2}\right)z; W_4 = \tfrac{(2.25)^2}{\Gamma(1+4\alpha)}\left(8 + \tfrac{2\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2} + \tfrac{4\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)}\right)W_0; \ldots
\end{aligned}
\tag{22}
$$

Please cite this article in press as: Singh BK, Kumar P, FRDTM for numerical simulation of multi-dimensional, time-fractional model of Navier–Stokes equation, Ain Shams Eng J (2016), http://dx.doi.org/10.1016/j.asej.2016.04.009

By using inverse FRDT, we have

$$\begin{aligned}
u(x,y,z,t) &= U_0 + U_1 t^{\alpha} + U_2 t^{2\alpha} + U_3 t^{3\alpha} + U_4 t^{4\alpha} + \ldots \\
&= -0.5x + y + z - \tfrac{2.25}{\Gamma(1+\alpha)} x t^{\alpha} + \tfrac{2(2.25)}{\Gamma(1+2\alpha)}(-0.5x + y + z)t^{2\alpha} - \tfrac{(2.25)^2}{\Gamma(1+3\alpha)}\left(4 + \tfrac{\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2}\right) x t^{3\alpha} \\
&\quad + \tfrac{(2.25)^2}{\Gamma(1+4\alpha)}\left(8 + \tfrac{2\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2} + \tfrac{4\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)}\right)(-0.5x + y + z)t^{4\alpha} + \ldots \\
v(x,y,z,t) &= V_0 + V_1 t^{\alpha} + V_2 t^{2\alpha} + V_3 t^{3\alpha} + V_4 t^{4\alpha} + \ldots \\
&= x - 0.5y + z - \tfrac{2.25}{\Gamma(1+\alpha)} y t^{\alpha} + \tfrac{2(2.25)}{\Gamma(1+2\alpha)}(x - 0.5y + z)t^{2\alpha} - \tfrac{(2.25)^2}{\Gamma(1+3\alpha)}\left(4 + \tfrac{\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2}\right) y t^{3\alpha} \\
&\quad + \tfrac{(2.25)^2}{\Gamma(1+4\alpha)}\left(8 + \tfrac{2\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2} + \tfrac{4\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)}\right)(x - 0.5y + z)t^{4\alpha} + \ldots \\
w(x,y,z,t) &= W_0 + W_1 t^{\alpha} + W_2 t^{2\alpha} + W_3 t^{3\alpha} + W_4 t^{4\alpha} + \ldots \\
&= x + y - 0.5z - \tfrac{2.25}{\Gamma(1+\alpha)} z t^{\alpha} + \tfrac{2(2.25)}{\Gamma(1+2\alpha)}(x + y - 0.5z)t^{2\alpha} - \tfrac{(2.25)^2}{\Gamma(1+3\alpha)}\left(4 + \tfrac{\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2}\right) z t^{3\alpha} \\
&\quad + \tfrac{(2.25)^2}{\Gamma(1+4\alpha)}\left(8 + \tfrac{2\Gamma(1+2\alpha)}{(\Gamma(1+\alpha))^2} + \tfrac{4\Gamma(1+3\alpha)}{\Gamma(1+\alpha)\Gamma(1+2\alpha)}\right)(x + y - 0.5z)t^{4\alpha} + \ldots
\end{aligned}$$

which is the required exact solution. For $\alpha = 1$, we have

$$\left.\begin{aligned}
u(x,y,z,t) &= (-0.5x + y + z)(1 + 2.25t^2 + 2.25^2 t^4 + \ldots) - 2.25xt(1 + 2.25t^2 + \ldots) \\
&= \tfrac{-0.5x+y+z-2.25xt}{1-2.25t^2}. \\
v(x,y,z,t) &= (x - 0.5y + z)(1 + 2.25t^2 + 2.25^2 t^4 + \ldots) - 2.25yt(1 + 2.25t^2 + \ldots) \\
&= \tfrac{x-0.5y+z-2.25yt}{1-2.25t^2}. \\
w(x,y,z,t) &= (x + y - 0.5z)(1 + 2.25t^2 + 2.25^2 t^4 + \ldots) - 2.25zt(1 + 2.25t^2 + \ldots) \\
&= \tfrac{x+y-0.5z-2.25zt}{1-2.25t^2}.
\end{aligned}\right\} \quad (23)$$

which is the exact solution of the associated classical NS equation for the velocity field which is same as reported in [42]. The velocity profile $(u, v, w)$ of the Navier–Stokes equation for $\alpha = 1$ is depicted in Fig. 7.

## 4. Conclusion

In this paper, `fractional reduced differential transformation method` is adopted for the numerical simulation of time-fractional model of Navier–Stokes equations with initial conditions. The fractional derivative is considered in the Caputo sense. The analytical results have been given in terms of a power series. Three test problems are carried out in order to validate and illustrate the efficiency of the method. The proposed solutions agree excellently with HPM [15] and ADM [16], and are approximated without any discretization, transformation, perturbation, or restrictive conditions. However, the performed calculations show that the described method needs very small size of computation in comparison with HPM [15] and ADM [16]. Small size of computation contrary to the other schemes, is the strength of the scheme.

## Acknowledgments

The authors are grateful to the anonymous referees for their time, effort, and extensive comments which improve the quality of the presentation of the paper. Pramod Kumar is also thankful to Babasaheb Bhimrao Ambedkar University, Lucknow, India, for financial assistance to carry out the work.

## References

[1] Podlubny I. Fractional differential equations. San Diego: Academic Press; 1999.
[2] Miller KS, Ross B. An Introduction to the fractional calculus and fractional differential equations. New York: Wiley; 1993.
[3] Caputo M, Mainardi F. Linear models of dissipation in anelastic solids. Rivist Nuovo Cimento 1971;1:161–98.
[4] Carpinteri A, Mainardi F. Fractals and fractional calculus in continuum mechanics. Wien (New York): Springer-Verlag; 1997.
[5] Hilfer R. Applications of fractional calculus in physics. Singapore: World Scientific; 2000.
[6] Klafter J, Lim SC, Metzler R. Fractional dynamics: recent advances. World Scientific; 2012.
[7] Baleanu D, Machado JAT, Luo AC. Fractional dynamics and control. Springer; 2012.
[8] Goldfain E. Fractional dynamics, cantorian space-time and the gauge hierarchy problem. Chaos Solitons Fract 2004;22 (3):513–20.
[9] He JH. Nonlinear oscillation with fractional derivative and its applications. In: Int conf vibrating Engg'98, Dalian. p. 288–91.
[10] He JH. Homotopy perturbation technique. Comput Methods Appl Mech Eng 1999;178:257–62.
[11] He JH. Approximate analytical solution for seepage flow with fractional derivatives in porous media. Comput Methods Appl Mech Eng 1998;167:57–68.
[12] Sushila, Singh J, Shishodia YS. A new reliable approach for two-dimensional and axisymmetric unsteady flows between parallel plates. Z Naturforsch A 2013;68a:629–34.

[13] Singh J, Kumar D, Kiliçman A. Numerical solutions of nonlinear fractional partial differential equations arising in spatial diffusion of biological populations. Abstr Appl Anal 2014;2014. Article ID 535793, 12 pages.
[14] Ragab AA, Hemida KM, Mohamed MS, Abd El Salam MA. Solution of time-fractional Navier–Stokes equation by using homotopy analysis method. Gen Math Notes 2012;13(2):13–21.
[15] Ganji ZZ, Ganji DD, Ganji AD, Rostamian M. Analytical solution of time-fractional Navier–Stokes equation in polar coordinate by homotopy perturbation method. Numer Meth Part Diff Eq 2010;26(1):117–24.
[16] Birajdar GA. Numerical solution of time fractional Navier–Stokes equation by discrete Adomian decomposition method. Nonlinear Eng 2014;3(1):21–6.
[17] Momani S, Odibat Z. Analytical solution of a time-fractional Navier–Stokes equation by Adomian decomposition method. Appl Math Comput 2006;177:488–94.
[18] Kumar D, Singh J, Kumar S. Numerical computation of nonlinear fractional Zakharov–Kuznetsov equation arising in ion-acoustic waves. J Egypt Math Soc 2014;22(3):373–8.
[19] Keskin Y, Oturanc G. Reduced differential transform method for partial differential equations. Int J Nonlinear Sci Numer Simul 2009;10(6):741–9.
[20] Keskin Y, Oturanc G. Reduced differential transform method: a new approach to factional partial differential equations. Nonlinear Sci Lett A 2010;1:61–72.
[21] Abazari R, Ganji M. Extended two-dimensional DTM and its application on nonlinear PDEs with proportional delay. Int J Comput Math 2011;88(8):1749–62.
[22] Gupta PK. Approximate analytical solutions of fractional Benney–Lin equation by reduced differential transform method and the homotopy perturbation method. Comput Math Appl 2011;58:2829–42.
[23] Abazari R, Abazari M. Numerical simulation of generalized Hirota–Satsuma coupled KdV equation by RDTM and comparison with DTM. Commun Nonlinear Sci Numer Simul 2012;17:619–29.
[24] Srivastava VK, Mishra N, Kumar S, Singh BK, Awasthi MK. Reduced differential transform method for solving $(1+n)$-dimensional Burgers' equation. Egypt J Basic Appl Sci 2014;1:115–9.
[25] Srivastava VK, Kumar S, Awasthi MK, Singh BK. Two-dimensional time fractional-order biological population model and its analytical solution. Egypt J Basic Appl Sci 2014;1:71–6.
[26] Singh BK, Srivastava VK. Approximate series solution of multi-dimensional, time fractional-order (heat-like) diffusion equations using FRDTM. R Soc Open Sci 2, 140511. doi:http://dx.doi.org/10.1098/rsos.140511.
[27] Srivastava VK, Awasthi MK, Tamsir M. RDTM solution of Caputo time fractional-order hyperbolic telegraph equation. AIP Adv 2013;3:032142.
[28] Srivastava VK, Awasthi MK, Chaurasia RK, Tamsir M. The telegraph equation and its solution by Reduced Differential Transform Method. Model Simul Eng 2013;2013:1–6.
[29] Shukla HS, Tamsir Mohammad, Srivastava VK, Kumar Jai. Approximate analytical solution of time-fractional order Cauchy–Reaction diffusion equation. CMES 2014;103(1):1–17.
[30] Dhiman Neeraj, Chauhan Anand. An approximate analytical solution description of time-fractional order Fokker–Plank equation by using FRDTM. Asia Pacific J Eng Sci Technol 2015;1(1):34–47.
[31] Shukla HS, Malik Gufran. Biological population model and its solution by reduced differential transform method. Asia Pacific J Eng Sci Technol 2015;1(1):1–10.
[32] Tamsir M, Srivastava VK. Analytical study of time-fractional order Klein–Gordon equation. Alexandria Eng J 2016. http://dx.doi.org/10.1016/j.aej.2016.01.025.
[33] Saravanan A, Magesh N. A comparison between the reduced differential transform method and the Adomian decomposition method for the Newell–Whitehead–Segel equation. J Egypt Math Soc 2013;21(3):259–65.
[34] Saha Ray S. A new coupled fractional reduced differential transform method for the numerical solutions of 2 dimensional time fractional coupled Burger equations. Model Simul Eng 2014;2014. Article ID 960241, 12 pages.
[35] Magesh N, Saravanan A. The reduced differential transform method for solving the systems of two dimensional nonlinear Volterra integro – differential equations. In: Proceedings of the international conference on mathematical sciences (ICMS-2014). Elsevier. p. 217–220. ISBN-978-93-5107-261-4.
[36] Saravanan A, Magesh N. An efficient computational technique for solving the Fokker–Planck equation with space and time fractional derivatives. J King Saud Univ – Sci; in press. doi:http://dx.doi.org/10.1016/j.jksus.2015.01.003.
[37] Navier CLM H. Memoire sur les lois du mouvement des fluides. Mem Acad Sci Inst France 1822;6:389–440.
[38] El-Shahed M, Salem A. On the generalized Navier–Stokes equations. Appl Math Comput 2005;156(1):287–93.
[39] Kumar D, Singh J, Kumar S. A fractional model of Navier–Stokes equation arising in unsteady flow of a viscous fluid. J Assoc Arab Univ Basic Appl Sci 2015;17:14–9.
[40] Kumar S, Kumar D, Abbasbandy S, Rashidi MM. Analytical solution of fractional Navier–Stokes equation by using modified Laplace decomposition method. Ain Shams Eng J 2014;5(2):569–74.
[41] Chaurasia VBL, Kumar D. Solution of the time-fractional Navier–Stokes equation. Gen Math Notes 2011;4(2):49–59.
[42] Campos MD, Romão EC. A high-order finite-difference scheme with a linearization technique for solving of three-dimensional Burgers equation. Comput Model Eng Sci 2014;103(3):139–54.

**Brajesh Kumar Singh** has completed Ph.D. in Cryptography from Indian Institute of Technology Roorkee (2012). He worked as Assistant Professor in the Department of Mathematics, Graphics Era Hill University, Dehradun, India from August 2012 to March 2015. Currently, he is working as Assistant Professor in the Department of Applied Mathematics, in Babasaheb Bhimrao Ambedkar University Lucknow INDIA. His research interest is in the area of Applied Mathematics such as Discrete Mathematics, Numerical Analysis, Numerical Solutions to Partial Differential Equations, Mathematical Modeling, Computational Fluid Dynamics, Computational aspects in Physics, Biology and Finance, etc.

**Pramod Kumar** is a research scholar in the Department of Applied Mathematics, in BBA University, Lucknow, India. His research interest is in Numerical Simulation of Mathematical Models.

Please cite this article in press as: Singh BK, Kumar P, FRDTM for numerical simulation of multi-dimensional, time-fractional model of Navier–Stokes equation, Ain Shams Eng J (2016), http://dx.doi.org/10.1016/j.asej.2016.04.009

Ain Shams Engineering Journal xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Ain Shams Engineering Journal

journal homepage: www.sciencedirect.com

# A note on solving the fourth-order Kuramoto-Sivashinsky equation by the compact finite difference scheme

Brajesh Kumar Singh [a,*], Geeta Arora [b], Pramod Kumar [a]

[a] Department of Applied Mathematics, School for Physical Sciences, Babasaheb Bhimrao Ambedkar University, Lucknow 226025, India
[b] Department of Mathematics, Lovely Professional University, Phagwara 144411, Punjab, India

ARTICLE INFO

*Article history:*
Received 26 February 2016
Revised 23 October 2016
Accepted 9 November 2016
Available online xxxx

*Keywords:*
Kuramoto-Sivashinsky equation
Compact finite difference scheme
SSP-RK43 scheme
Thomas algorithm

ABSTRACT

The present article is concerned with the implementation of the compact finite difference scheme, in the space and the optimal four-stage, order three strong stability-preserving time-stepping Runge-Kutta (SSP-RK43) scheme, in time for computation of one dimensional *Kuramoto-Sivashinsky* equation (KSE), arises in the study of flame front propagation, phase turbulence in reaction-diffusion system and in many other biological and chemical processes. The efficiency of proposed scheme is confirmed by six test problems with known exact solutions. The numerical results demonstrate the reliability and efficiency of the algorithm developed.



## 1. Introduction

The *Kuramoto-Sivashinsky* equation was originally derived in the context of plasma instabilities, flame front propagation, and phase turbulence in reaction-diffusion system [40]. This equation models the fluctuations of the position of a flame front, the motion of a fluid going down a vertical wall, or a spatially uniform oscillating chemical reaction in a homogeneous medium [3]. It exhibits chaotic behavior, having solution like traveling waves moving without change of shape over a finite spatial domain. It also has a wide range of applications in a variety of physical phenomena such as reaction diffusion systems [23], long waves on the interface between two viscous fluids [16], hydrodynamics thin films [50], and flame front instability [46].

This paper is concerned with the numerical simulation of one-dimensional *Kuramoto-Sivashinsky* equation

$$u_t + uu_x + \alpha u_{xx} + \gamma u_{xxxx} = 0, \tag{1}$$

subject to the initial condition $(*): u(x,0) = g(x),\ x \in \mathbb{R}$ and the boundary conditions $(**)$: $u(a,t) = \psi_1(t),\quad u(b,t) = \psi_2(t),$ $u_x(a,t) = u_x(b,t) = u_{xx}(a,t) = u_{xx}(b,t) = 0$, where $\mathbb{R}$ is the set of real numbers, $f$, $g$, $\psi_1$, $\psi_2$ are known functions, and $u_t = \frac{\partial u}{\partial t}$, $u_{xx} = \frac{\partial^2 u}{\partial x^2}$, etc. The nonlinear term in KSE counterbalance the dispersion term while dissipation terms shows a mechanism for energy transfer.

In the recent years, various type of numerical schemes have been developed for numerical simulation of time dependent partial differential equations [28,29,26,5–7]. The KSE has been studied numerically by several schemes: Chebyshev spectral collocation methods [24], local discontinuous Galerkin methods [28], tanh function method [27], Homotopy analysis method [25], the inverse scattering method [13], homogeneous balance method [14], cubic B-spline finite difference-collocation method [26], Quintic B-spline collocation method (QBSC) [38], Septic B-spline collocation method (SBSC) [30], A higher-order finite element approach [31], Finite difference discretization [33], A fourth-order singly diagonally implicit Runge-Kutta method [35], Lattice Boltzmann Method (LBM) [29]. For more schemes, we refer the readers to [34,32,9,8–10] and the references therein.

The compact finite difference scheme (CFDS) has been implemented for numerical simulation of various types of partial differential equations [36] such as hyperbolic equations [54], Navier-Stokes equations [42,51], reaction-diffusion [53], equation and Schrödinger equation [12]. A sixth-order compact finite difference scheme has been used for the numerical solution of Helmholtz equation [49], advection diffusion equation [15], integro-differential equations [55], Burgers' equation [43], Burger-Fisher equation [45], Fisher's equation [2], and Sine-Gordon equation [44].

Peer review under responsibility of Ain Shams University.

ELSEVIER Production and hosting by Elsevier

* Corresponding author.
*E-mail addresses:* bksingh0584@gmail.com (B.K. Singh), geetadma@gmail.com (G. Arora), bbaupramod@gmail.com (P. Kumar).

http://dx.doi.org/10.1016/j.asej.2016.11.008
2090-4479/

In this paper, a new numeric solution of KSE is obtained by adopting CFDS. Using CFDS to KSE gets converted into a system of first-order system of ordinary differential equations (ODEs), in time. The resulting system of ODEs is solved by using SSP-RK43 scheme [48]. Six test problems considered by different researchers, have been studied to demonstrate the accuracy and utility of the proposed scheme.

The rest of the paper is organized as follows. Description of a compact finite difference scheme is presented in Section 2. Section 3 describes the procedure for implementation of the method. Six test problems are given to illustrate the applicability and accuracy of the proposed method in Section 4. Section 5 concludes the applicability and accuracy of the proposed schemes.

## 2. Description of a Compact Finite Difference Schemes (CFDS)

In order to get the numerical solution of KSE (1) by CFDS, the spatial and time discretization is needed. The computational domain $\mathbb{D} \subset \mathbb{R}$ defined by $\mathbb{D} = \{x \in \mathbb{R} : a \leqslant x \leqslant b\}$, is portioned uniformly with the following knots:

$$a = x_1 < x_2, \ldots, x_{N-1} < x_N = b$$

with $h = \frac{b-a}{N-1}$.

The procedure of finding the spatial derivatives using CFDS is as follows:

### 2.1. First-order spatial derivative

The first-order derivative $i$-th collocation point can be given as

$$\theta u'_{i-1} + u'_i + \theta u'_{i+1} = b\frac{u_{i+2} - u_{i-2}}{4h} + a\frac{u_{i+1} - u_{i-1}}{2h} \tag{2}$$

In particular, for $a = \frac{2}{3}(2+\theta)$, $b = \frac{1}{3}(4\theta - 1)$, Eq. (2) get reduced into a $\theta$-family of fourth-order tridiagonal schemes. Further, for $\theta = \frac{1}{3}$ the above scheme (2) is reduced into the *sixth-order tridiagonal scheme* as

$$u'_{i-1} + 3u'_i + u'_{i+1} = \frac{1}{12h}(u_{i+2} + 28u_{i+1} - 28u_{i-1} - u_{i-2}), \tag{3}$$

with the truncation error $\frac{4}{7!}h^6 u_i^{(7)}$ [36]. The approximation for the derivatives at the boundary points $x_1$, $x_2$, $x_{N-1}$ and $x_N$, derived from one-sided schemes [43], respectively are

$$u'_1 + 5u'_2 = \frac{1}{60h}(-197u_1 - 25u_2 + 300u_3 - 100u_4 + 25u_5 - 3u_6)$$

$$2u'_1 + 11u'_2 + 2u'_3 = \frac{1}{12h}(-80u_1 - 35u_2 + 136u_3 - 28u_4 + 8u_5 - u_6)$$

$$2u'_{N-2} + 11u'_{N-1} + 2u'_N = \frac{1}{12h}(80u_N + 35u_{N-1} - 136u_{N-2} + 28u_{N-3} - 8u_{N-4} + u_{N-5})$$

$$5u'_{N-1} + u'_N = \frac{1}{60h}(197u_N + 25u_{N-1} - 300u_{N-2} + 100u_{N-3} - 25u_{N-4} + 3u_{N-5})$$

The matrix representation of the scheme is given as

$$\begin{bmatrix} 1 & 5 & & & & & \\ 2 & 11 & 2 & & & & \\ & 1 & 3 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & 3 & 1 & \\ & & & & 2 & 11 & 2 \\ & & & & & 5 & 1 \end{bmatrix} \begin{bmatrix} u'_1 \\ u'_2 \\ \vdots \\ u'_i \\ \vdots \\ u'_{N-1} \\ u'_N \end{bmatrix} = \begin{bmatrix} \psi_1(u) \\ \psi_2(u) \\ \vdots \\ \psi_i(u) \\ \vdots \\ \psi_{N-1}(u) \\ \psi_N(u) \end{bmatrix} \tag{4}$$

where $u = (u_1, u_2, \ldots, u_N)^T$ and

$$\psi_1(u) = \frac{1}{60h}(-197u_1 - 25u_2 + 300u_3 - 100u_4 + 25u_5 - 3u_6)$$

$$\psi_2(u) = \frac{1}{12h}(-80u_1 - 35u_2 + 136u_3 - 28u_4 + 8u_5 - u_6)$$

$$\psi_i(u) = \frac{1}{12h}(u_{i+2} + 28u_{i+1} - 28u_{i-1} - u_{i-2}), \qquad i = 3, 4, \ldots, N-2,$$

$$\psi_{N-1}(u) = \frac{1}{12h}(80u_N + 35u_{N-1} - 136u_{N-2} + 28u_{N-3} - 8u_{N-4} + u_{N-5}), \text{ and}$$

$$\psi_N(u) = \frac{1}{60h}(197u_N + 25u_{N-1} - 300u_{N-2} + 100u_{N-3} - 25u_{N-4} + 3u_{N-5})$$

### 2.2. Higher-order spatial derivative

The second-order derivative at $i$th collocation point can be given as follow

$$\theta u''_{i-1} + u''_i + \theta u''_{i+1} = b\frac{u_{i+2} - 2u_i + u_{i-2}}{4h^2} + a\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}. \tag{5}$$

In particular, for $a = \frac{4}{3}(1-\theta)$, $b = \frac{1}{3}(-1 + 10\theta)$, Eq. (5) get reduced into a $\theta$-family of fourth-order tridiagonal scheme. For $\theta = \frac{1}{10}$, Eq. (5) reduced to

$$u''_{i-1} + 10u''_i + u''_{i+1} = \frac{12}{h^2}(u_{i+1} - 2u_i + u_{i-1}) \tag{6}$$

At the boundary nodes, $x_1$ and $x_N$, the approximation formulae for the second-order derivatives are derived from Taylor series expansion about the nodes [2,55] as follows:

$$10u''_1 + u''_2 = \frac{12}{h^2}\left(\frac{115}{36}u_1 - \frac{1555}{144}u_2 + \frac{89}{6}u_3 - \frac{773}{72}u_4 + \frac{151}{36}u_5 - \frac{11}{16}u_6\right)$$

$$u''_{N-1} + 10u''_N = \frac{12}{h^2}\left(-\frac{11}{16}u_{N-5} + \frac{151}{36}u_{N-4} - \frac{773}{72}u_{N-3} + \frac{89}{6}u_{N-2} - \frac{1555}{144}u_{N-1} + \frac{115}{36}u_N\right)$$

The above scheme for second order approximation can be written in matrix form as follows

$$\begin{bmatrix} 10 & 1 & & & & \\ 1 & 10 & 1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & 1 & 10 & 1 & \\ & & & 1 & 10 & \end{bmatrix} \begin{bmatrix} u''_1 \\ u''_2 \\ \vdots \\ u''_{N-1} \\ u''_N \end{bmatrix} = \begin{bmatrix} \phi_1(u) \\ \phi_2(u) \\ \vdots \\ \phi_{N-1}(u) \\ \phi_N(u) \end{bmatrix} \tag{7}$$

where

$$\phi_1(u) = \frac{12}{h^2}\left(\frac{115}{36}u_1 - \frac{1555}{144}u_2 + \frac{89}{6}u_3 - \frac{773}{72}u_4 + \frac{151}{36}u_5 - \frac{11}{16}u_6\right)$$

$$\phi_i(u) = \frac{12}{h^2}(u_{i+1} - 2u_i + u_{i-1}), \qquad i = 2, 3, \ldots, N-1, \text{ and}$$

$$\phi_N(u) = \frac{12}{h^2}\left(-\frac{11}{16}u_{N-5} + \frac{151}{36}u_{N-4} - \frac{773}{72}u_{N-3} + \frac{89}{6}u_{N-2} - \frac{1555}{144}u_{N-1} + \frac{115}{36}u_N\right).$$

Please cite this article in press as: Singh BK et al. A note on solving the fourth-order Kuramoto-Sivashinsky equation by the compact finite difference scheme. Ain Shams Eng J (2016), http://dx.doi.org/10.1016/j.asej.2016.11.008

The fourth-order spatial derivative is obtained by replacing $u$ by $u''$ in (7):

$$Au^{iv} = \phi(u''), \tag{8}$$

where $\phi(u'') = (\phi_1(u''), \phi_2(u''), \ldots, \phi_N(u''))$ and $A$ is the coefficient matrix defined in Eq. (7). The spatial derivatives at each of the nodal points are obtained by solving the above tridiagonal matrix system of linear equations by the well-known "Thomas algorithm".

## 3. Implementation of the method

On putting the values of spatial derivatives approximated by CFDS, Eq. (1) is reduced into a system of first-order ODEs:

$$\frac{du_i}{dt} = L(u_i), \quad i \in \{1, \ldots, N\}, \tag{9}$$

with initial condition $(*)$ and boundary conditions $(**)$, where $L \rightarrow$ nonlinear differential operator defined by

$$L(u_i) = -u(x_i)u'(x_i) - \alpha u''(x_i) - \gamma u^{iv}(x_i).$$

Eq. (9) is solved by SSP-RK43 scheme through the following steps:

$$u^{(1)} = u^m + \frac{\triangle t}{2} L(u^m),$$

$$u^{(2)} = u^{(1)} + \frac{\triangle t}{2} L(u^{(1)}),$$

$$u^{(3)} = \frac{2}{3}u^m + \frac{u^{(2)}}{3} + \frac{\triangle t}{6} L(u^{(2)}),$$

$$u^{m+1} = u^{(3)} + \frac{\triangle t}{2} L(u^{(3)}).$$

and thus, the solutions of $u(x,t)$ at the required time level are obtained.

## 4. Numerical experiments and discussions

This section presents the numerical computation of six problems by adopting CFDS. The accuracy and efficiency of the method is measured by evaluating the $L_2$, $L_\infty$ and the global relative (GRE) errors norms for the test problems whose analytical solutions exist.

$$L_2 = \left( h \sum_{j=1}^{N} \left[ u_j - u_j^* \right]^2 \right)^{\frac{1}{2}}, \quad L_\infty = \max_{j=1}^{N} \left| u_j - u_j^* \right|,$$
$$\mathrm{GRE} = \frac{\left( \sum_j |u_j - u_j^*| \right)}{\left( \sum_j |u_j^*| \right)}, \tag{10}$$

where $u_j$ and $u_j^*$, respectively denote the numerical solution and the analytical solution at the node $j$.

**Example 1.** Consider the KSE (1) with $\alpha = -1$ and $\gamma = 1$ as

$$u_t + uu_x - u_{xx} + u_{xxxx} = 0, \qquad t > 0.$$

The initial and boundary conditions are extracted form the exact solution (11):

$$u(x,t) = \beta + \frac{15\tanh^3(k(x - \beta t - x_0)) - 45\tanh(k(x - \beta t - x_0))}{19^{3/2}}. \tag{11}$$

The numerical solution is obtained for $\beta = 5$, $k = \frac{1}{2\sqrt{19}}$ and $x_0 = -25$ with $N = 101, 141$ and time-step $\triangle t = 0.1, 0.01$. The comparison of the GRE obtained by CFDS at different time levels $t \leqslant 12$ taking $\triangle t = 0.01$ and $N = 101$ with the errors obtained by the earlier schemes: LBM [17], QBSC [38] and SBSC [30], is reported in Table 1. The $L_\infty$ and the global relative errors at $t \in \{1, 2, 3, 4, 5\}$ are compared with the errors obtained by SBSC [30] in Table 3. The errors are reported in Tables 4 and 2 taking different number of nodal points and time-steps. It is evident that the proposed numerical results are more accurate than that obtained in SBSC [30], LBM [17] and QBSC [38]. The absolute errors and the physical behavior of the CFDS solutions at different time levels are depicted in Fig. 1.

**Example 2.** Consider the KSE (1) with $\alpha = 1$ and $\gamma = 1$ as

$$u_t + uu_x + u_{xx} + u_{xxxx} = 0, \qquad t > 0.$$

The initial and boundary conditions are extracted form the exact solution (12):

$$u(x,t) = \beta + \frac{15}{19} \times \sqrt{\frac{11}{19}} \left[ -9\tanh(k(x - \beta t - x_0)) + 11\tanh^3(k(x - \beta t - x_0)) \right]. \tag{12}$$

The numerical solution is obtained for $\beta = 5$, $k = 0.5\sqrt{\frac{11}{19}}$ and $x_0 = -12$. The comparison of the GREs obtained by CFDS at different time levels $t \leqslant 4$ taking $\triangle t = 0.01$ and $N = 63$ with the errors obtained by the earlier schemes: LBM [17] and QBSC [38] is presented in Table 5. It is evident that the proposed numerical results are comparable to the results that obtained in LBM [17] and QBSC [38]. The numeric results with the analytical solutions for the computational domain $[-30, 30]$, are depicted at different time levels, in Fig. 2.

**Example 3.** The KSE (1) with $\alpha = -\nu$ and $\gamma = 0$ get reduced into Burgers' equation (@) $u_t + uu_x - \nu u_{xx} = 0$, $x \in [0, 1.2]$, subject to the initial condition: $u(x, t = 1) = x\left[1 + \exp\frac{1}{4\nu}x^2 - \frac{1}{4}\right]^{-1}$, and the boundary conditions $u(0,t) = u(1.2,t) = 0$, $t > 1$. The exact solution of the equation is

$$u(x,t) = \frac{x}{t}\left[1 + \frac{t}{t_0}^{1/2} \exp\frac{x^2}{4\nu t}\right]^{-1}, \qquad t_0 = \exp 1/8\nu, \quad \text{for } t \geqslant 1.$$

The comparison of the CFDS numeric solutions at different time levels, are presented in Table 6 for selected node points, with parameters $\nu = 0.005$, $h = 0.01$ and $\triangle t = 0.01$. The comparison is done with solutions obtained by Mittal and Jain [39], Shu et al. [47], Arora-Singh [1], and the exact solutions. In Table 7, $L_2$ and $L_\infty$ errors for $\nu = 0.005$ at different time levels $t \leqslant 3.5$ are compared with the errors obtained by several earlier schemes. At $t = 3.6$, $L_2$ and $L_\infty$ errors are $10^{-5}$ and $7 \times 10^{-5}$, respectively which are same as in

**Table 1**
Comparison of GRE at different time levels $t \leqslant 12$ with the errors due to well known earlier schemes [38], LBM [17], and SBSC [30] with $\triangle t = 0.01$.

| $t$ | QBSC $N = 200$ | LBM $N = 200$ | SBSC $N = 200$ | CFDS $N = 100$ |
|---|---|---|---|---|
| 6 | 6.50927E−06 | 7.8808E−06 | 1.62464E−07 | 8.43458E−08 |
| 8 | 7.13154E−06 | 9.5324E−06 | 1.94032E−07 | 8.91175E−08 |
| 10 | 7.31029E−06 | 1.0891E−05 | 2.22878E−07 | 9.25444E−08 |
| 12 | 8.77659E−06 | 1.1793E−05 | 5.31428E−07 | 1.69186E−07 |

Please cite this article in press as: Singh BK et al. A note on solving the fourth-order Kuramoto-Sivashinsky equation by the compact finite difference scheme. Ain Shams Eng J (2016), http://dx.doi.org/10.1016/j.asej.2016.11.008

**Table 2**
Comparison of the $L_\infty$ errors at time $t = 1$ and $t = 2$ taking different number of nodal pints and $\triangle t = 0.01$.

| $N$ | 15 | 30 | 60 | 120 |
|---|---|---|---|---|
| $t = 1$ | 2.98E−01 | 9.83E−04 | 1.61E−05 | 8.20E−07 |
| $t = 2$ | 9.33E−01 | 5.96E−03 | 2.45E−05 | 1.32E−06 |

**Table 3**
Comparison of global relative and $L_\infty$ errors at different time levels, $t \leqslant 5$ with the errors due to SBSC [30] with parameter $N = 141$, $\triangle t = 0.01$.

| Scheme | Errors | $t = 1$ | $t = 2$ | $t = 3$ | $t = 4$ | $t = 5$ |
|---|---|---|---|---|---|---|
| CFDS | GRE | 9.5984E−09 | 1.4368E−08 | 1.7535E−08 | 1.9749E−08 | 2.2235E−08 |
| | $L_\infty$ | 4.6014E−07 | 6.6612E−07 | 8.1212E−07 | 9.0924E−07 | 9.7622E−07 |
| SBSC | GRE | 2.9543E−08 | 4.9686E−08 | 6.7009E−08 | 8.2345E−08 | 9.6082E−08 |
| | $L_\infty$ | 3.1791E−06 | 5.9019E−06 | 8.2807E−06 | 1.0385E−05 | 1.2264E−05 |

**Table 4**
Comparison of the errors at different time levels $t \leqslant 12$.

| $N : \triangle t$ | | $t = 1$ | $t = 2$ | $t = 3$ | $t = 4$ | $t = 6$ | $t = 8$ | $t = 10$ | $t = 12$ |
|---|---|---|---|---|---|---|---|---|---|
| 101 : 0.01 | $L_\infty$ | 1.727E−06 | 2.685E−06 | 3.283E−06 | 3.681E−06 | 4.153E−06 | 4.399E−06 | 4.534E−06 | 3.040E−05 |
| | $L_2$ | 3.961E−06 | 6.265E−06 | 7.764E−06 | 8.791E−06 | 1.005E−05 | 1.073E−05 | 1.114E−05 | 3.356E−05 |
| | GRE | 3.415E−08 | 5.358E−08 | 6.609E−08 | 7.438E−08 | 8.435E−08 | 8.912E−08 | 9.254E−08 | 1.692E−07 |
| 101 : 0.1 | $L_\infty$ | 1.097E−05 | 1.990E−05 | 2.727E−05 | 3.343E−05 | 4.306E−05 | 5.014E−05 | 5.545E−05 | 2.052E−04 |
| | $L_2$ | 2.997E−05 | 5.451E−05 | 7.504E−05 | 9.240E−05 | 1.199E−04 | 1.404E−04 | 1.559E−04 | 2.709E−04 |
| | GRE | 2.824E−07 | 5.050E−07 | 6.886E−07 | 8.440E−07 | 1.087E−06 | 1.266E−06 | 1.401E−06 | 1.939E−06 |
| 51 : 0.1 | $L_\infty$ | 3.029E−05 | 5.117E−05 | 5.532E−05 | 6.668E−05 | 7.263E−05 | 7.497E−05 | 7.577E−05 | 7.614E−05 |
| | $L_2$ | 7.465E−05 | 1.147E−04 | 1.391E−04 | 1.550E−04 | 1.740E−04 | 1.849E−04 | 1.920E−04 | 1.971E−04 |
| | GRE | 6.759E−07 | 9.793E−07 | 1.255E−06 | 1.367E−06 | 1.540E−06 | 1.629E−06 | 1.693E−06 | 1.634E−06 |

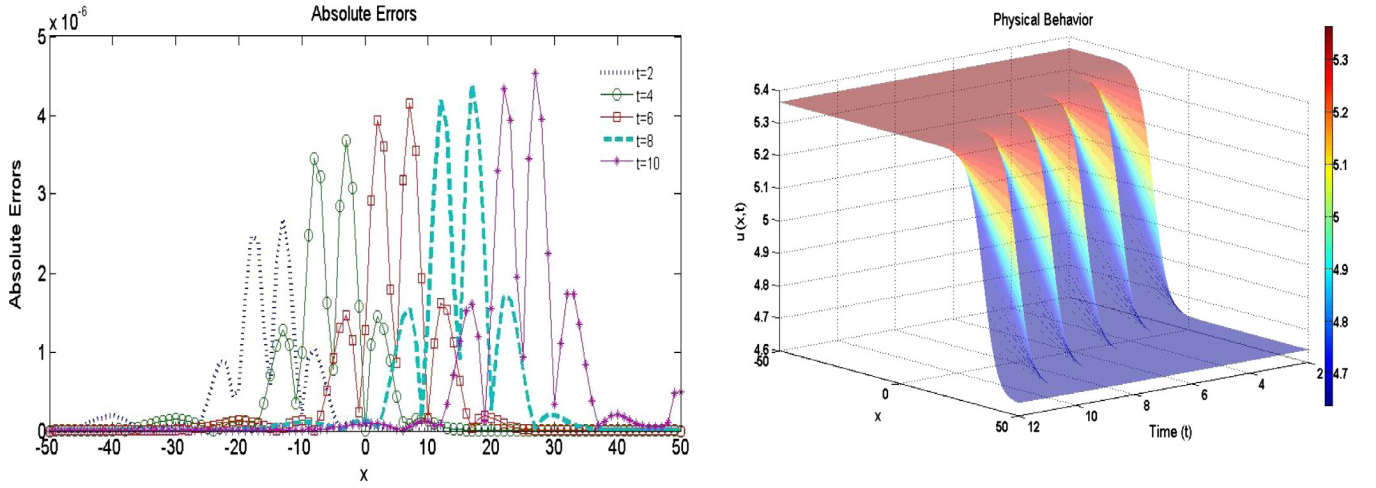


**Figure 1.** The absolute errors and the physical behavior of Example 1 in $[-50, 50]$ with $\triangle t = 0.01$ and $N = 101$ at different time levels $t \leqslant 10$.

**Table 5**
Comparison of GRE at different time levels $\leqslant 4$ with the errors in LBM [17] and QBSC [38], for $\triangle t = 0.01$.

| Scheme | $t = 1$ | $t = 2$ | $t = 3$ | $t = 4$ |
|---|---|---|---|---|
| QBSC ($N$ = 150) | 3.82E−04 | 5.51E−04 | 7.04E−04 | 8.64E−04 |
| LBM ($N$ = 150) | 6.79E−04 | 1.15E−03 | 1.59E−03 | 2.01E−03 |
| CFDS ($N$ = 63) | 4.33E−04 | 5.79E−04 | 8.74E−04 | 3.57E−03 |

MCB-DQM [1] and are very less in comparison to the errors obtained by the three methods proposed in [22]. Thus, CFDS produces more accurate numeric solutions which are better than almost all the earlier schemes, and approaching towards the exact solutions. The absolute errors and physical behavior are depicted in Fig. 3.

**Example 4.** Consider the KSE (1) with $\alpha = 1$, and $\gamma = 0$, i.e., Burgers' equation (@) in the domain $[0, 1]$ with initial condition $u(x, 0) = \sin(\pi x)$, and boundary conditions $u(0, t) = u(1, t) = 0$.

The analytical solution of this problem is given by

$$u(x,t) = \frac{4\pi\nu\sum_{j=1}^{\infty} j I_j \frac{1}{2\pi\nu} \sin j\pi x exp - j^2\pi^2\nu t}{I_0 \frac{1}{2\pi\nu} + 2\sum_{j=1}^{\infty} I_j \frac{1}{2\pi\nu} \cos j\pi x exp - j^2\pi^2\nu t}, \tag{13}$$

where $I_j$ are the modified Bessel's functions.

The numerical solutions are obtained for different values of $\nu$ as follows:

Please cite this article in press as: Singh BK et al. A note on solving the fourth-order Kuramoto-Sivashinsky equation by the compact finite difference scheme. Ain Shams Eng J (2016), http://dx.doi.org/10.1016/j.asej.2016.11.008

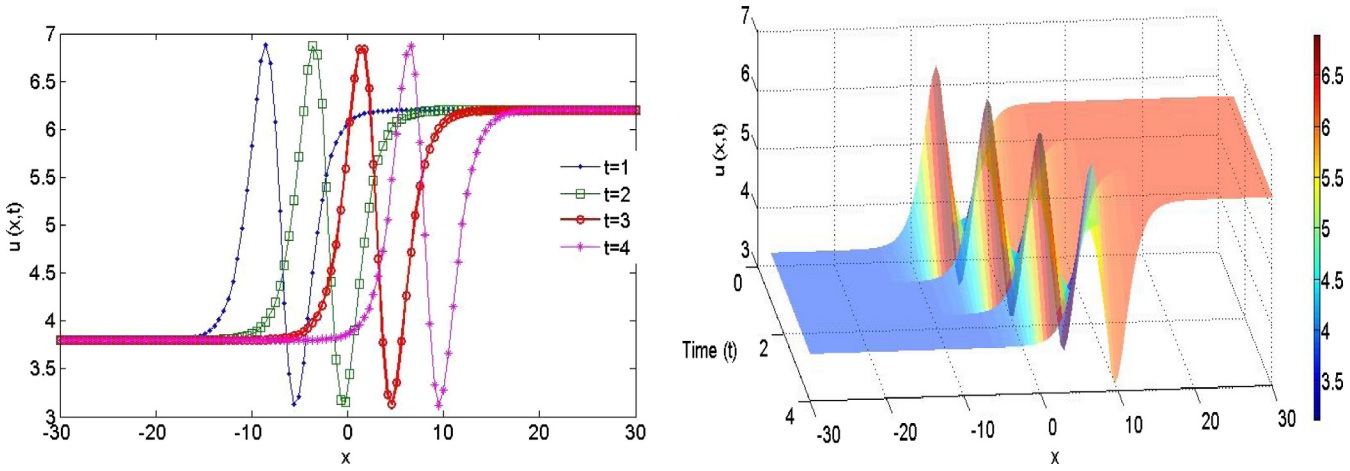


**Figure 2.** The physical behavior of Example 2 in $[-30, 30]$ with $\triangle t = 0.01$, $N = 63$ at different time levels in two and three-dimension $t \leqslant 4$.

**Table 6**
Comparison of the CFDS solutions of Example 3 with exact solutions, for $\nu = 0.005$.

| $x$ | $t$ | Shu et al. [47] with $h = 10^{-4}$ | | Mittal and Jain [39] | MCB-DQM [1] | CFDS | Exact value |
|---|---|---|---|---|---|---|---|
| | | $\beta = 1$ $\triangle t = 0.01$ | $\beta = 0.5$ $\triangle t = 0.01$ | $h = 0.005$ $\triangle t = 10^{-3}$ | $h = 0.01$ $\triangle t = 0.01$ | $h = 0.01$ $\triangle t = 0.01$ | |
| 0.20 | 1.7 | 0.1176565 | 0.1174841 | 0.1176452 | 0.1176450 | 0.1176450 | 0.1176452 |
| | 2.5 | 0.0800527 | 0.0798389 | 0.0799990 | 0.0799989 | 0.0799990 | 0.0799990 |
| | 3.0 | 0.0667147 | 0.0665176 | 0.0666658 | 0.0666658 | 0.0666658 | 0.0666658 |
| | 3.5 | 0.0571820 | 0.0570060 | 0.0571422 | 0.0571422 | 0.0571422 | 0.0571422 |
| 0.40 | 1.7 | 0.2332111 | 0.2348504 | 0.2351690 | 0.2351680 | 0.2351680 | 0.2351677 |
| | 2.5 | 0.1591735 | 0.1596608 | 0.1599771 | 0.1599770 | 0.1599770 | 0.1599769 |
| | 3.0 | 0.1328314 | 0.1330273 | 0.1333211 | 0.1333210 | 0.1333210 | 0.1333209 |
| | 3.5 | 0.1139606 | 0.1140077 | 0.1142780 | 0.1142780 | 0.1142780 | 0.1142779 |
| 0.6 | 1.7 | 0.2940048 | 0.2961269 | 0.2958570 | 0.2959160 | 0.2959090 | 0.2959097 |
| | 2.5 | 0.2347876 | 0.2376699 | 0.2381299 | 0.2381200 | 0.2381210 | 0.2381207 |
| | 3.0 | 0.1973222 | 0.1990478 | 0.1994839 | 0.1994800 | 0.1994810 | 0.1994805 |
| | 3.5 | 0.1697753 | 0.1708231 | 0.1712257 | 0.1712240 | 0.1712240 | 0.1712242 |
| 0.8 | 1.7 | 0.0008917 | 0.0006640 | 0.0006381 | 0.0006464 | 0.0006464 | 0.0006465 |
| | 2.5 | 0.1103866 | 0.1036067 | 0.1021325 | 0.1020930 | 0.1020940 | 0.1020957 |
| | 3.0 | 0.2088346 | 0.2093735 | 0.2088032 | 0.2088380 | 0.2088360 | 0.2088359 |
| | 3.5 | 0.2119293 | 0.2143409 | 0.2145938 | 0.2145870 | 0.2145870 | 0.2145869 |

**Table 7**
Comparison of $L_2$ and $L_\infty$ errors in the CFDS solutions of Example 3 for $\nu = 0.005$ at different time levels $t \leqslant 3.5$.

| Methods | $N$ | $\triangle t$ | $t = 1.7$ | | $t = 2.4$ | | $t = 3.1$ | | $t = 3.25$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | $L_2 \times 10^3$ | $L_\infty \times 10^3$ | $L_2 \times 10^3$ | $L_\infty \times 10^3$ | $L_2 \times 10^3$ | $L_\infty \times 10^3$ | $L_2 \times 10^3$ | $L_\infty \times 10^3$ |
| CFDS | 121 | 0.01 | 0.00095 | 0.00428 | 0.00041 | 0.00167 | 0.00051 | 0.00330 | 0.001284 | 0.009148 |
| MCB-DQM [1] | 121 | 0.01 | 0.00191 | 0.00777 | 0.00086 | 0.00308 | 0.00065 | 0.00331 | 0.001341 | 0.00918 |
| QRTDQ [20] | 101 | 0.001 | 0.109 | 0.434 | 0.100 | 0.339 | 0.091 | 0.266 | | |
| CBCDQ [19] | 101 | 0.001 | | 0.210 | 0.680 | 0.190 | 0.530 | | | |
| BS.FEM [4] | 50 | 0.1 | 0.857 | 2.576 | 0.423 | 1.242 | 0.230 | 0.680 | | |
| C.S.C. [41] | 50 | 0.1 | 0.857 | 2.576 | 0.423 | 1.242 | 0.235 | 0.688 | | |
| Galerkin [56] | 200 | 0.01 | 0.857 | 2.576 | 0.423 | 1.242 | 0.235 | 0.688 | | |
| QBCM1 [37] | 200 | 0.01 | 0.017 | 0.061 | 0.012 | 0.058 | 0.601 | 4.434 | | |
| QBCM2 [37] | 200 | 0.01 | 0.358 | 1.211 | 0.251 | 0.807 | 0.630 | 4.790 | | |
| PDQ [18] | 200 | 0.01 | 0.015 | 0.056 | 0.011 | 0.064 | 0.584 | 4.301 | | |
| | | | | | $t = 2.5$ | | | | | |
| QBCM [11] | 200 | 0.01 | 0.072 | 0.311 | 0.051 | 0.189 | | | 1.129 | 8.983 |
| CBCM [11] | 200 | 0.01 | 2.466 | 27.577 | 2.111 | 25.15 | | | 1.925 | 21.084 |
| QRKM [11] | 200 | 0.01 | 0.026 | 0.091 | 0.031 | 0.115 | | | 1.111 | 8.000 |
| | | | | | | | $t = 3.00$ | | $t = 3.50$ | |
| CFDS | 121 | 0.01 | 0.00095 | 0.00428 | 0.00038 | 0.00152 | 0.00033 | 0.00158 | 0.005815 | 0.04090 |
| MCB-DQM [1] | 121 | 0.01 | 0.00191 | 0.00777 | 0.00778 | 0.00275 | 0.00056 | 0.0017 | 0.006177 | 0.04335 |
| MCB-CM [39] | 241 | 0.01 | 0.0252 | 0.0994 | 0.0151 | 0.0549 | 0.0118 | 0.0414 | 0.0117 | 0.0486 |
| [47] ($\beta = 0.5$) | 12001 | 0.01 | 0.38421 | 1.34728 | 0.49135 | 1.55470 | 0.51508 | 1.5529 | 0.525855 | 1.52196 |
| [47] ($\beta = 1$) | 12001 | 0.01 | 3.08966 | 10.4040 | 2.72048 | 8.29747 | 2.39922 | 6.9880 | 2.12110 | 5.94321 |

Please cite this article in press as: Singh BK et al. A note on solving the fourth-order Kuramoto-Sivashinsky equation by the compact finite difference scheme. Ain Shams Eng J (2016), http://dx.doi.org/10.1016/j.asej.2016.11.008

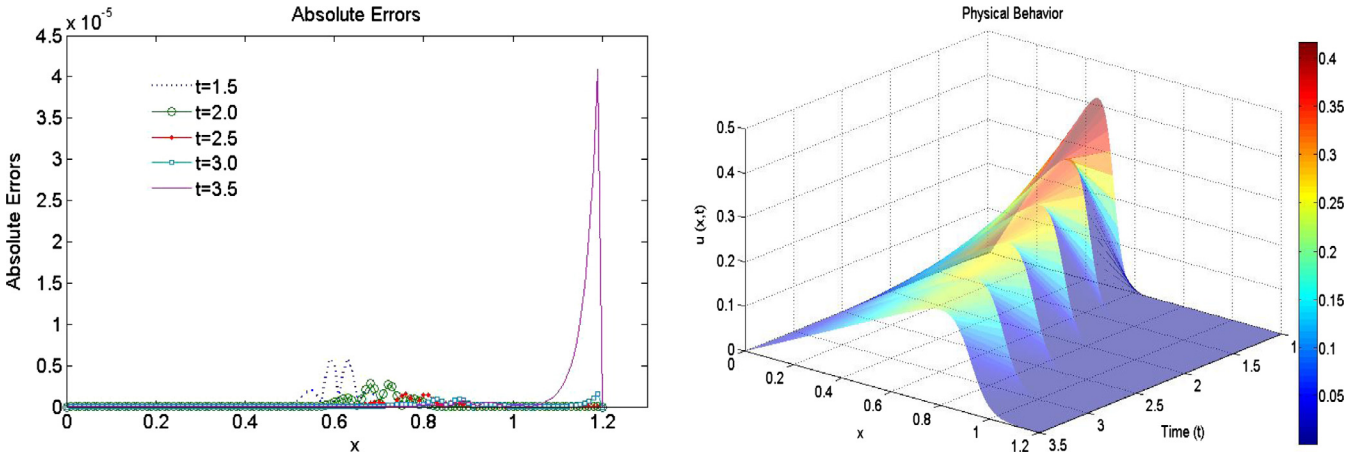


**Figure 3.** The absolute errors and the physical behavior of Example 3 at different time levels $t \leqslant 3.5$.

**Table 8**
Comparison of CFDS solutions of Example 4 for $v = 1.0$ with earlier schemes, and exact solutions with parameter $h = 0.05$ and $\triangle t = 0.001$.

| $x$ | | [11] | [39] | [21] | [1] | [38] | CFDS | Exact |
|---|---|---|---|---|---|---|---|---|
| | $\triangle t \rightarrow$ | $10^{-4}$ | 0.00025 | 0.000125 | 0.00025 | 0.001 | 0.001 | |
| | $t$ | 0.0125 | 0.025 | 0.025 | 0.025 | 0.004 | 0.05 | $\leftarrow h$ |
| 0.25 | 0.4 | 0.01357 | 0.01354 | 0.01363 | 0.0135710 | ... | ... | 0.01357 |
| | 0.6 | 0.00189 | 0.00188 | 0.00190 | 0.0018888 | 0.00185 | 0.001889 | 0.00189 |
| | 0.8 | 0.00026 | 0.00026 | 0.00026 | 0.0002624 | 0.00026 | 0.000262 | 0.00026 |
| | 1.0 | 0.00004 | 0.00004 | 0.00003 | 0.0000365 | 0.00004 | 0.000036 | 0.00004 |
| 0.50 | 0.4 | 0.01923 | 0.01920 | 0.01932 | 0.0192336 | ... | ... | 0.01923 |
| | 0.6 | 0.00267 | 0.00266 | 0.00269 | 0.0026719 | 0.00262 | 0.002672 | 0.00267 |
| | 0.8 | 0.00037 | 0.00037 | 0.00037 | 0.0003712 | 0.00036 | 0.000371 | 0.00037 |
| | 1.0 | 0.00005 | 0.00005 | 0.00005 | 0.0000516 | 0.00005 | 0.000051 | 0.00005 |
| 0.75 | 0.4 | 0.01362 | 0.01360 | 0.01369 | 0.0136298 | ... | ... | 0.01363 |
| | 0.6 | 0.00189 | 0.00188 | 0.00190 | 0.0018899 | 0.00185 | 0.001890 | 0.00189 |
| | 0.8 | 0.00026 | 0.00026 | 0.00026 | 0.0002625 | 0.00026 | 0.000262 | 0.00026 |
| | 1.0 | 0.00004 | 0.00004 | 0.00003 | 0.0000365 | 0.00004 | 0.000036 | 0.00004 |

**Table 9**
Comparison of CFDS solutions of Example 4 for $v = 0.1$ with solutions by earlier schemes, and exact solutions.

| $x$ | | [11] | [39] | [21] | [1] | [38] | CFDS | [43] | Exact |
|---|---|---|---|---|---|---|---|---|---|
| | $\triangle t \rightarrow$ | $10^{-4}$ | 0.0025 | 0.00125 | 0.004 | 0.0001 | 0.001 | 0.00001 | |
| | $t$ | 0.0125 | 0.025 | 0.025 | 0.025 | 0.004 | 0.025 | 0.0125 | $\leftarrow h$ |
| 0.25 | 0.4 | 0.30890 | 0.30892 | 0.30910 | 0.3089280 | – | 0.308894 | 0.308894 | 0.30889 |
| | 0.6 | 0.24075 | 0.24077 | 0.24093 | 0.2407550 | 0.24073 | 0.240739 | 0.240739 | 0.24074 |
| | 0.8 | 0.19569 | 0.19572 | 0.19586 | 0.1956840 | 0.19519 | 0.195676 | 0.195676 | 0.19568 |
| | 1.0 | 0.16258 | 0.16261 | 0.16274 | 0.1625700 | 0.16206 | 0.162565 | 0.162565 | 0.16256 |
| | 3.0 | 0.02720 | 0.02718 | 0.02720 | 0.0272047 | 0.02699 | 0.027202 | – | 0.02720 |
| 0.50 | 0.4 | 0.56965 | 0.56970 | 0.56973 | 0.5696530 | – | 0.569629 | 0.569632 | 0.56963 |
| | 0.6 | 0.44723 | 0.44729 | 0.44736 | 0.4472170 | 0.44683 | 0.447205 | 0.447206 | 0.44721 |
| | 0.8 | 0.35925 | 0.35930 | 0.35943 | 0.3592450 | 0.35875 | 0.359235 | 0.359236 | 0.35924 |
| | 1.0 | 0.29192 | 0.29195 | 0.29213 | 0.2919250 | 0.29136 | 0.291915 | 0.291916 | 0.29192 |
| | 3.0 | 0.04019 | 0.04016 | 0.04032 | 0.0402085 | 0.03993 | 0.040205 | – | 0.04021 |
| 0.75 | 0.4 | 0.62538 | 0.62520 | 0.62573 | 0.6253490 | – | 0.625361 | 0.625438 | 0.62544 |
| | 0.6 | 0.48715 | 0.48694 | 0.48760 | 0.4872040 | 0.48613 | 0.487212 | 0.487215 | 0.48721 |
| | 0.8 | 0.37385 | 0.37365 | 0.37434 | 0.3739350 | 0.37281 | 0.373920 | 0.373922 | 0.37392 |
| | 1.0 | 0.28741 | 0.28724 | 0.28788 | 0.2874930 | 0.28642 | 0.287473 | 0.287474 | 0.28747 |
| | 3.0 | 0.02976 | 0.02974 | 0.29881 | 0.0297753 | 0.02952 | 0.029772 | – | 0.02977 |

(a) The numerical solutions are computed at different time levels with parameter values $v = 1.0$, $h = 0.05$ and $\triangle t = 0.001$ at some selected node points. Table 8 presents the comparison of CFDS numeric solutions with the exact solutions and the solutions due to schemes in [1,11,39,38,21]. It is found that CFDS produces comparable results.

(b) The numerical solutions are computed at different time levels with parameter values $v = 0.1$, $h = 0.025$ and $\triangle t = 0.004$, presented in Table 9. It is evident that the CFDS solutions are comparably much better to the numeric solution due to earlier schemes [1,43,11,38,39,21], and approaching towards exact solutions.

Please cite this article in press as: Singh BK et al. A note on solving the fourth-order Kuramoto-Sivashinsky equation by the compact finite difference scheme. Ain Shams Eng J (2016), http://dx.doi.org/10.1016/j.asej.2016.11.008

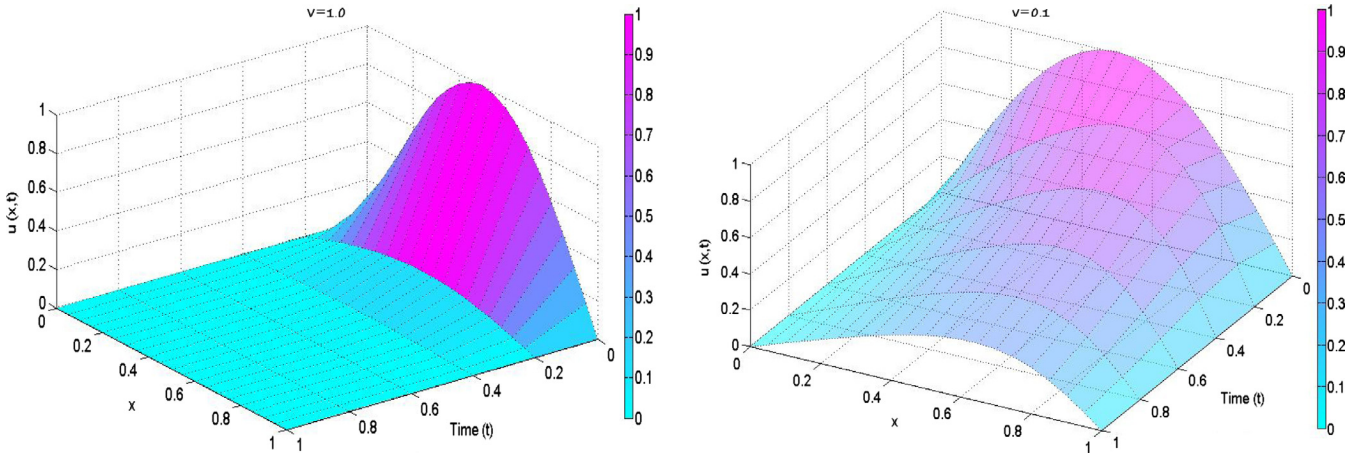


**Figure 4.** Physical behavior of CFSDS6 numeric solutions of Example 4 for $v = 1.0$ and $v = 0.1$ at $t \leqslant 1$ with $N = 21$, $\triangle t = 0.001$.

**Table 10**
Comparison of the errors in Example 5 for $t = 1$ with the parameter $\triangle t = 0.01$ and $N = 40$.

| Errors | CFDS | Lakestani and Dehghan [26] | |
|---|---|---|---|
| | | $(j = 3)$ | $(j = 4)$ |
| $L_\infty$ | 8.58E−04 | 4.80E−03 | 2.10E−03 |
| $L_2$ | 1.44E−03 | 4.40E−03 | 1.60E−03 |

**Table 11**
The errors in Example 5 at different time levels $t \leqslant 5$ with the parameters $\triangle t = 0.01$ and $N = 40$.

| Errors | $t = 0.5$ | $t = 1$ | $t = 2$ | $t = 3$ | $t = 4$ | $t = 5$ |
|---|---|---|---|---|---|---|
| $L_\infty$ | 8.07E−04 | 8.58E−04 | 1.01E−03 | 1.24E−03 | 1.31E−03 | 1.32E−03 |
| $L_2$ | 1.29E−03 | 1.44E−03 | 1.94E−03 | 2.45E−03 | 2.84E−03 | 3.01E−03 |
| GrE | 1.17E−04 | 1.46E−04 | 2.19E−04 | 3.11E−04 | 3.97E−04 | 4.25E−04 |

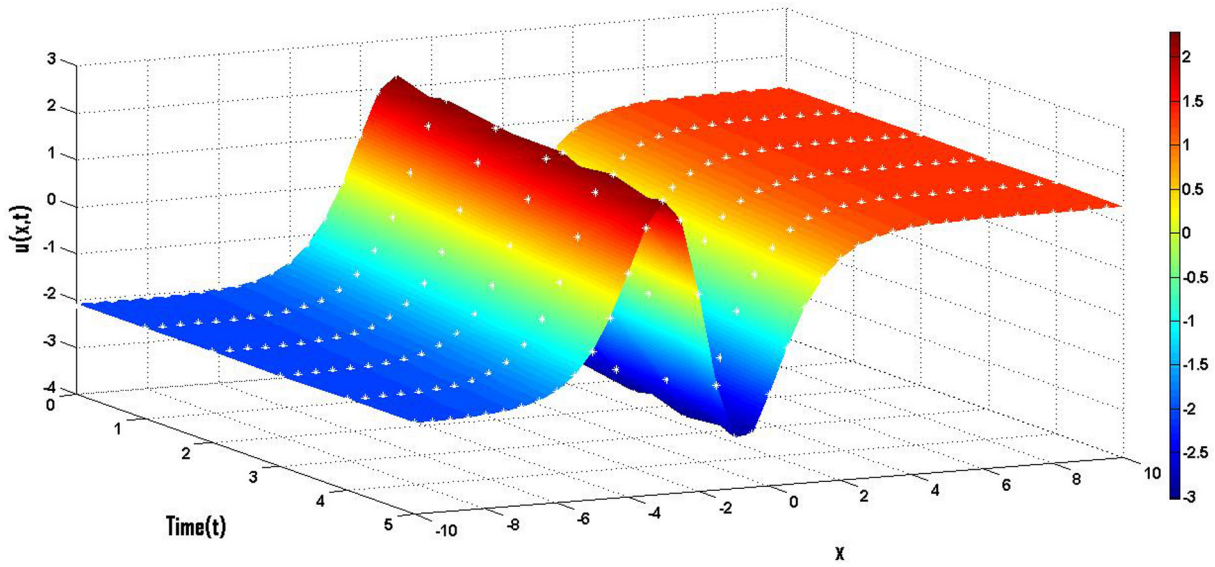


**Figure 5.** Physical behavior of numeric solutions of Example 5 at $t \leqslant 5$ with $N = 41$, $\triangle t = 0.01$.

The physical behavior of the solution for $v = 0.1, 1.0$ is depicted in Fig. 4.

**Example 5.** Consider the KSE (1) with $\alpha = 1$ and $\gamma = 0.5$ in the domain $[-10, 10]$. The solution is obtained for the initial and boundary conditions extracted from the exact solutions, as given below

$$u(x,t) = -\frac{0.1}{k} + \frac{60}{19}k\left(\alpha - 38\gamma k^2\right)\tanh(\theta) + 120\gamma k^3\tanh^3(\theta), \quad (14)$$

where $\theta = kx + 0.1t$ and $k = \frac{1}{2}\sqrt{\frac{11\alpha}{19\gamma}}$.

The errors are found in Table 10 are comparable with that the errors obtained by Lakestani and Dehghan's [26] method based on the finite difference and collocation method using B-splines. In Table 11, the GRE, $L_2$ and $L_\infty$ errors have been computed for

Please cite this article in press as: Singh BK et al. A note on solving the fourth-order Kuramoto-Sivashinsky equation by the compact finite difference scheme. Ain Shams Eng J (2016), http://dx.doi.org/10.1016/j.asej.2016.11.008

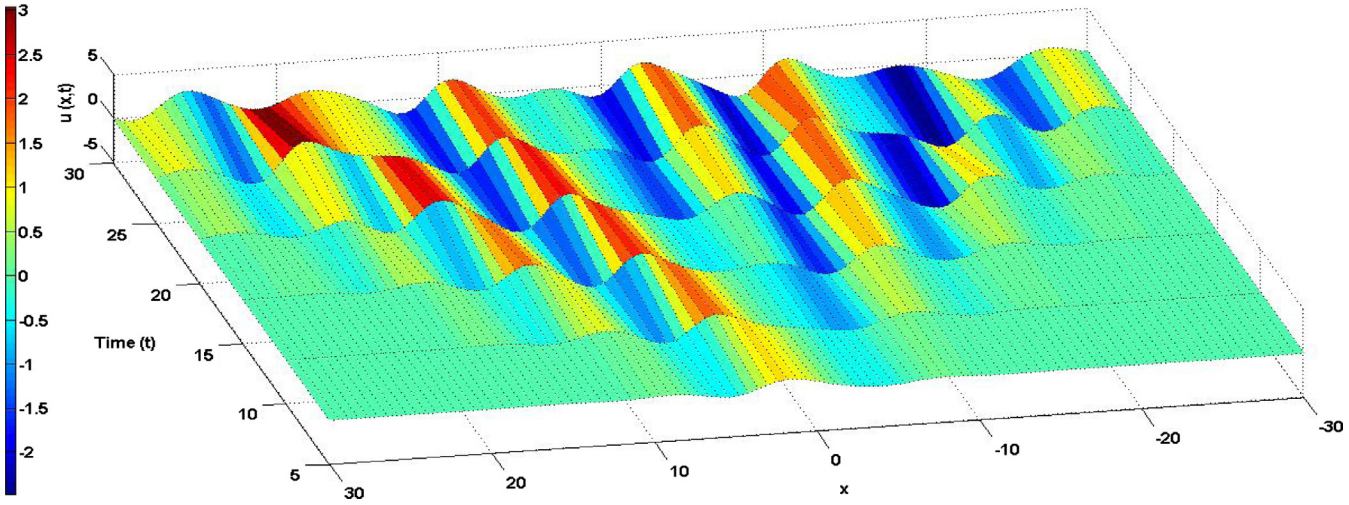


**Figure 6.** Physical behavior of numeric solutions of Example 6 at $t \leqslant 5$ with $N = 101$, $\triangle t = 0.01$.

different time levels $0.5 \leqslant t \leqslant 5$ with $\triangle t = 0.01$, $N = 40$. The numerical solution are also presented graphically in Fig. 5 at different time levels $1 \leqslant t \leqslant 5$, which shows the same characteristics as depicted in [26].

**Example 6.** Consider the KSE (1) with $\alpha = 1$ and $\gamma = 1$ in the domain $[-30, 30]$ which results in the nonlinear partial differential equation exhibiting the chaotic behavior over a finite spatial domain. The solution is obtained for the Gaussian initial condition [52,38,26]: $u(x, 0) = \exp(-x^2)$ with the BCs $u(-30, t) = u(30, t) = 0$.

The numerical solutions at different time levels $5 \leqslant t \leqslant 30$ are presented graphically in Fig. 6, and it is found that the solutions depicts the same characteristics as in it were in [52,38,26]).

## 5. Conclusion

This paper present a new numerical simulation of *Kuramoto-Sivashinsky* equation which is approximated by using the compact finite difference scheme (CFDS). CFDS is used to convert KSE into a system of first-order ODEs, in time and then SSP-RK43 scheme is used to solve the resulting system of ODEs. Six test problems considered by different researchers have been studied to confirm the accuracy and utility of the proposed scheme. The findings can be summarized as follows.

(a) The proposed results are found to be in good agreement with the exact solutions. The effects of space and time resolutions on the accuracy of the present scheme are also investigated through detailed simulations.
(b) The advantage of this scheme is that it is easy to implement to a (non) linear equation and easy to program. It require less number of grid points, so it leads to the least accumulation of numerical errors.
(c) The approximate solutions have been computed without any transformation or linearization. Easy, computational efficiency and economical implementation is the strength of this scheme.
(d) This scheme can be adopted as a promising technique to solve multidimensional problems arising in various engineering and physical modeling problems.

## Acknowledgement

The authors would like to thank the anonymous referees for their time, effort, and extensive comments on the revision of the manuscript which improve the quality of the presentation of the paper.

## References

[1] Arora G, Singh BK. Numerical solution of Burgers' equation with modified cubic B-spline differential quadrature method. Appl Math Comput 2013;224:166–77.
[2] Bastani M, Salkuyeh DK. A highly accurate method to solve Fisher's equation. Pramana - J Phys Ind Acad Sci 2012;78(3):335–46.
[3] Conte R. Exact solutions of nonlinear partial differential equations by singularity analysis. Lecture notes in physics, vol. 632. Springer; 2003. p. 183.
[4] Chung J, Kim E, Kim Y. Asymptotic agreement of moments and higher order contraction in the Burgers equation. J Diff Eqn 2010;248(10):2417–34.
[5] Arora G, Mittal RC, Singh BK. Numerical solution of BBM-burger equation with quartic B-spline collocation method. J Eng Sci Technol 2014;(Special Issue 1):104–16.
[6] Srivastava VK, Singh BK. A Robust finite difference scheme for the numerical solutions of two dimensional time-dependent coupled nonlinear Burgers' equations. Int J Appl Math Mech 2014;10(7):28–39.
[7] Singh BK, Arora G. A numerical scheme to solve Fisher-type reaction-diffusion equations. Nonlin Stud/MESA- Math Eng, Sci Aerosp 2014;5(2):153–64.
[8] Singh BK, Arora G, Singh MK. A numerical scheme for the generalized Burgers-Huxley equation. J Egypt Math Soc 2016, http://dx.doi.org/10.1016/j.joems.2015.11.003.
[9] Singh BK, Bianca C. A new numerical approach for the solutions of partial differential equations in three-dimensional space. Appl Math Inf Sci 2016;10(5):1–10.
[10] Singh BK, Kumar P. A novel approach for numerical computation of Burgers' equation $(1 + 1)$ and $(2 + 1)$ dimension. Alex Eng J 2016, http://dx.doi.org/10.1016/j.aej.2016.08.023.
[11] Dağ İ, Irk D, Sahin A. B-Spline collocation methods for numerical solutions of the Burgers' equation. Math Prob Eng 2005;5:521–38.
[12] Dehghan M, Taleei A. A compact split-step finite difference method for solving the nonlinear Schrodinger equations with constant and variable coefficients. Comp Phys Commun 2010;181(1):43–51.
[13] Drain PG, Johnson RS. Solitons: an introduction. New York: Cambridge University Press; 1989.
[14] Fan EG, Zhang HQ. A note on the homogeneous balance method. Phys Lett A 1998;264:403–6.
[15] Gürarslan G, Karahan H, Alkaya D, Sari M, Yasar M. Numerical solution of advection-diffusion equation using a sixth-order compact finite difference method. Math Prob Eng 2013;2013:7. http://dx.doi.org/10.1155/2013/672936672936.
[16] Hooper AP, Grimshaw R. Nonlinear instability at the interface between two viscous fluids. Phys Fluids 1985;28:37–45.
[17] Huilin L, Changfeng M. Lattice Boltzmann method for the generalized KuramotoSivashinsky equation. Physica A 2009;388:14051412.
[18] Korkmaz A. Numerical solutions of some nonlinear partial differential equations using Differential Quadrature Method, Thesis of Master Degree. Eskişehir Osmangazi University; 2006.

Please cite this article in press as: Singh BK et al. A note on solving the fourth-order Kuramoto-Sivashinsky equation by the compact finite difference scheme. Ain Shams Eng J (2016), http://dx.doi.org/10.1016/j.asej.2016.11.008

[19] Korkmaz A. Numerical Solutions of some one dimensional partial differential equations using B-spline Differential Quadrature Method, Doctoral Dissertation. Eskişehir Osmangazi University; 2010.
[20] Korkmaz A, Aksoy AM, Dağ İ. Quartic B-spline differential quadrature method. Int J Nonlin Sci 2011;11(4):403–11.
[21] Korkmaz A, Dağ İ. Shock wave simulations using sinc differential quadrature method. Eng Comput: Int J Comp-Aided Eng Soft 2011;28(6):654–74.
[22] Korkmaz A, Dağ İ. Cubic B-spline differential quadrature methods and stability for Burgers' equation. Eng Comput: Int J Comp-Aided Eng Soft 2013;30 (3):320–44.
[23] Kuramoto Y, Tsuzuki T. Persistent propagation of concentration waves in dissipative media far from thermal equilibrium. Prog Theor Phys 1976;55:356–569.
[24] Khater AH, Temsah RS. Numerical solutions of the generalized Kuramoto-Sivashinsky equation by Chebyshev spectral collocation methods. Comput Math Appl 2008;56:1465–72.
[25] Kurulay M, Secer A, Akinlar MA. A new approximate analytical solution of Kuramoto-Sivashinsky equation using Homotopy analysis method. Appl Math Inf Sci 2013;7(1):267–71.
[26] Lakestania M, Dehghan M. Numerical solutions of the generalized Kuramoto-Sivashinsky equation using B-spline functions. Appl Math Model 2012;36 (2):605–17.
[27] Lan H, Wang K. Exact solutions for two nonlinear equations. J Phys A, Math Gen 1990;23:3923–8.
[28] Yan X, Shu CW. Local discontinuous Galerkin methods for the Kuramoto-Sivashinsky equations and the Ito-type coupled KdV equations. Comput Meth Appl Mech Eng 2006;195:3430–47.
[29] Ye L, Yan G, Li T. Numerical method based on the Lattice Boltzmann model for the Kuramoto-Sivashinsky equation. J Sci Comput 2011;2011. http://dx.doi.org/10.1007/s10915-010-9455-1.
[30] Zarebnia M, Parvaz R. Septic B-spline collocation method for numerical solution of the Kuramoto-Sivashinsky equation. Int J Math Comput Sci Eng 2013;2(1):55–61.
[31] Anders D, Dittmann M, Weinberg K. A higher-order finite element approach to the Kuramoto-Sivashinsky equation. ZAMM Z Angew Math Mech 2012;92:599–607.
[32] Hesch C, SchuB S, Dittmann M, Franke M, Weinberg K. Isogeometric analysis and hierarchical refinement for higher-order phase-field models. Comput Meth Appl Mech Eng 2016;303:185–207.
[33] Akrivis G. Finite difference discretization of the Kuramoto-Sivashinsky equation. Numer Math 1992;63:1–11.
[34] Gomez H, París J. Numerical simulation of asymptotic states of the damped Kuramoto-Sivashinsky equation. Phys Rev E 2011;93:046702.
[35] Deng D, Pan T. A fourth-order singly diagonally implicit Runge-Kutta method for solving one-dimensional Burgers' equation. IAENG Int J Appl Math 2015;45 (4):327–33.
[36] Lele SK. Compact finite difference schemes with spectral-like resolution. J Comput Phys 1992;103(1):16–42.
[37] Lin S, Wang C, Dai Z. New exact traveling and non-traveling wave solutions for $(2+1)$ dimensional Burgers equation. Appl Math Comput 2010;216 (10):3105–10.
[38] Mittal RC, Arora G. Quintic B-spline collocation method for numerical solution of the Kuramoto-Sivashinsky equation. Commun Nonlin Sci Numer Simul 2010;15:2798–808.
[39] Mittal RC, Jain RK. Numerical solutions of nonlinear Burgers' equation with modified cubic B-splines collocation method. Appl Math Comput 2012;218 (2012):7839–55.
[40] Rademacher J, Wattenberg R. Viscous shocks in the destabilized Kuramoto-Sivashinsky equation. J Comput Nonlin Dyn 2006;1:336–47.
[41] Salas AH. Symbolic computation of solutions for a forced Burgers equation. Appl Math Comput 2010;216(1):18–26.
[42] Sanyasiraju YVSS, Manjula V. Higher order semi-compact scheme to solve transient incompressible Navier-Stokes equations. Comput Mech 2005;35 (6):441–8.
[43] Sari M, Gürarslan G. A sixth-order compact finite difference scheme to the numerical solutions of Burgers' equation. Appl Math Comput 2009;208 (2):475–83.
[44] Sari M, Gürarslan G. A sixth-order compact finite difference method for the one-dimensional sine-Gordon equation. Int J Numer Meth Biol Eng 2011;27 (7):1126–38.
[45] Sari M, Gürarslan G, Dağ I. A compact finite difference method for the solution of the generalized Burgers-Fisher equation. Numer Meth Partial Diff Eq 2010;26(1):125–34.
[46] Sivashinsky GI. Instabilities, pattern-formation, and turbulence in flames. Ann Rev Fluid Mech 1983;15:179–99.
[47] Shu-Sen Xie, Heo S, Kim S, Woo G, Yi S. Numerical solution of one-dimensional Burgers' equation using reproducing kernel function. J Comput Appl Math 2008;214:417–34.
[48] Spiteri JR, Ruuth SJ. A new class of optimal high-order strong stability-preserving time-stepping schemes. SIAM J Numer Anal 2002;40(2):469–91.
[49] Sutmann G. Compact finite difference schemes of sixth order for the Helmholtz equation. J Comput Appl Math 2007;203(1):15–31.
[50] Tadmor E. The well-posedness of the Kuramoto-Sivashinsky equation. SIAM J Math Anal 1986;17:884–93.
[51] Tian Z, Liang X, Yu P. A higher order compact finite difference algorithm for solving the incompressible Navier- Stokes equations. Int J Numer Meth Eng 2011;88(6):511–32.
[52] Uddin M, Sirajul H, Siraj-ul-Islam. A mesh-free numerical method for solution of the family of Kuramoto-Sivashinsky equations. Appl Math Comput 2009;212:45869.
[53] Wang YM, Zhang HB. Higher-order compact finite difference method for systems of reaction diffusion equations. J Comput Appl Math 2009;233:502–18.
[54] Wirz HJ, Schutter FD, Turi A. An implicit, compact, finite difference method to solve hyperbolic equations. Math Comput Simul 1977;19(4):241–61.
[55] Zhao J, Corless RM. Compact finite difference method for integro-differential equations. Appl Math Comput 2006;177(1):271–88.
[56] Zhang L, Ouyang J, Wang X, Zhang X. Variational multiscale element-free Galerkin method for 2*D* Burgers equation. J Comput Phys 2010;229 (19):7147–61.

**Brajesh Kumar Singh** has completed Ph.D. in Cryptography from Indian Institute of Technology Roorkee (2012). He worked as Assistant Professor, Department of Mathematics, Graphics Era Hill University, Dehradun INDIA from August 2012 to March 2015. Currently, he is working as Assistant Professor, Department of Applied Mathematics, Babasaheb Bhimrao Ambedkar University Lucknow INDIA. His research interest in area of Applied Mathematics such as Discrete Mathematics, Numerical Analysis, Numerical Solutions to Partial Differential Equations, Mathematical Modeling, Computational Fluid Dynamics, Computational aspects in Physics, Biology and Finance, etc.

**Geeta Arora** has completed Ph.D. in Numerical Analysis from Indian Institute of Technology Roorkee (2010). Currently, she is working as Assistant Professor, Department of Mathematics, LPU, Punjab, INDIA. Her research interest in area of Applied Mathematics such as Numerical Analysis, Numerical Solutions to Partial Differential Equations, Mathematical Modeling, etc.

**Pramod Kumar** is research scholar of Department of Applied Mathematics, BBA University Lucknow INDIA. His research interest is Numerical simulation of Mathematical models.

Please cite this article in press as: Singh BK et al. A note on solving the fourth-order Kuramoto-Sivashinsky equation by the compact finite difference scheme. Ain Shams Eng J (2016), http://dx.doi.org/10.1016/j.asej.2016.11.008