Journal of Mathematical Modeling
Vol. 11, No. 1, 2023, pp. 133–156. Research Article

JMM

An LN-stable method to solve the fractional partial integro-differential equations

Fahimeh Ziyaee, Abolfazl Tari[*]

Department of Mathematics, Shahed University, Tehran, Iran
Email(s): fahimeh.ziyaee@shahed.ac.ir, tari@shahed.ac.ir

Abstract. In this paper, a class of Volterra fractional partial integro-differential equations (VFPIDEs) with initial conditions is investigated. Here, the well-known method of lines (MOLs) is developed to solve the VFPIDEs. To this end, the VFPIDE is converted into a system of first-order ordinary differential equations (ODEs) in time variable with initial conditions. Then the resulting ODE system is solved by an LN-stable method, such as Radau IIA or Lobatto IIIC. It is proved that the proposed method is LN-stable. Also, the convergence of the proposed method is proved. Finally, some numerical examples are given to illustrate the efficiency and accuracy of the proposed method.

Keywords: Volterra fractional partial integro-differential equation, method of lines, LN-stability, convergence.
AMS Subject Classification 2010: 26A33, 65R20, 35R11, 45D05.

1 Introduction

Fractional calculus is one of the most innovative branches of mathematics which has attracted the interest of many researchers in recent decades [7]. It is a powerful tool in applied mathematics to study a wide range of problems from various fields of science and engineering, such as mathematical physics, finance, hydrology, biophysics, thermodynamics, control theory, statistical mechanics, astrophysics, cosmology and bioengineering [2]. Since fractional calculations and related modeling have received more attention from researchers and scientists in mathematics and other sciences, the solution of fractional differential and integro-differential equations have also been extensively studied.

The fractional integro-differential equation and their applications in heat conduction and electromagnetics can be found in [3, 23]. Up to now, many numerical methods have been proposed for solving fractional integro-differential equations in one and two-dimensional cases, for example: matrix-based

[*]Corresponding author.

Received: 30 July 2022 / Revised: 27 December 2022 / Accepted: 5 January 2023
DOI: 10.22124/JMM.2023.22727.2013

method [18], operational matrix method [11, 14], finite-difference Fibonacci collocation [12], Legendre wavelets method [15, 19], successive approximations method [20], expansion method [16], and the Legendre Lobatto collocation scheme [8].

In this paper, we consider the equation

$$D_{0t}^{\alpha}u(x,t)-\int_0^t\int_0^x k(x,t,y,z)u(y,z)dydz=f(x,t),\quad (x,t)\in[0,a]\times[0,b], \tag{1}$$

with $0<\alpha<1$ and the initial condition

$$u(x,0)=h_0(x), \tag{2}$$

where $f(x,t)$ and $k(x,t,y,z)$ are given continuous functions and $u(x,t)$ is unknown function. Also, D_{0t}^{α} denotes the time-fractional differential operator, of order α in the Caputo sense.

Here, we focus on the method of lines (MOLs) to solve (1)-(2), a general and well-established numerical procedure to solve partial differential equations (PDEs) [9, 21]. The main purpose of the MOLs is to replace the derivatives with respect to all variables except one in the PDE with algebraic approximations (by finite differences). Typically, the derivative with respect to time in a physical problem remains. In other words, only one independent variable remains, and the resulting semi-discrete problem is a system of coupled ordinary differential equations (ODEs) in time. Thus, to achieve an approximate solution of the PDE, we can apply an analytical or numerical algorithm to the resulting initial value ODEs. Therefore, we obtain a semi-discrete (semi-analytical) or fully discrete approximation of PDE. The semi-analytical formulation of the approximation leads to a straight algorithm with more accurate results than the other techniques. In the fully discrete formulation, stability and convergence are easily prepared.

2 Preliminaries

In this section, we give some basic definitions and concepts that we need in the following sections.

2.1 Runge-Kutta Methods

The ODEs according to dependence on time variable are classified into autonomous and non-autonomous systems. If a system of ODEs does not explicitly depend on the time-independent variable, it is called an autonomous system or a time-invariant system, otherwise it is called a non-autonomous system.

An autonomous and a non-autonomous system can be displayed as

$$\mathbf{y}'(t)=f(\mathbf{y}(t)),\quad t\in[t_0,b], \tag{3}$$

and

$$\mathbf{y}'(t)=f(t,\mathbf{y}(t)),\quad t\in[t_0,b], \tag{4}$$

respectively.

The Runge-Kutta (RK) methods are a set of implicit and explicit techniques designed to estimate the solution of ODEs. In general, the parameters of an m-stage implicit RK method are the weights

$\theta = (\theta_1, \theta_2, \ldots, \theta_m)^T \in \mathbb{R}^m$, abscissas (nodes) $\eta = (\eta_1, \eta_2, \ldots, \eta_m)^T \in \mathbb{R}^m$ and the coefficients matrix $\Lambda = (\lambda_{ij})_{i,j=1}^m \in \mathbb{R}^{m\times m}$ which are usually displayed in a table namely Butcher tableau as Table 1 [5]. An m-stage implicit RK method to solve the equation (4) with initial condition is displayed as the following general formula

$$R_{i,n} = \mathbf{y}_n + \Delta t \sum_{j=1}^{m} \lambda_{ij} f(t_n + \eta_j \Delta t, R_{j,n}), \quad i = 1, 2, \cdots, m, \tag{5}$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \sum_{j=1}^{m} \theta_j f(t_n + \eta_j \Delta t, R_{j,n}), \tag{6}$$

where $\Delta t = t_{n+l} - t_n$ is the step-size.

Table 1: Butcher tableau of RK method.

$$\begin{array}{c|cccc}
\eta_1 & \lambda_{11} & \lambda_{12} & \cdots & \lambda_{1m} \\
\eta_2 & \lambda_{21} & \lambda_{22} & \cdots & \lambda_{2m} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\eta_m & \lambda_{m1} & \lambda_{m2} & \cdots & \lambda_{mm} \\
\hline
 & \theta_1 & \theta_2 & \cdots & \theta_m
\end{array}
\qquad
\begin{array}{c|c}
\eta & \Lambda \\
\hline
 & \theta^T
\end{array}$$

2.2 AN-Stability and LN-Stability

Numerical stability is a notion and a favorable feature in the study of algorithms and methods used in the numerically solutions of problems. An algorithm or method is numerically stable if the amount of error during the solving process does not increase too much. There are different definitions of numerical stability, among which are AN-stability and LN-stability.

Let q(t) be a continuous complex-valued function with $\Re(q(t)) < 0$ for $t \in [t_0, b]$. Consider the non-autonomous test equation

$$\mathbf{y}' = q(t)\mathbf{y}, \quad q(t) \in \mathbb{R}. \tag{7}$$

Implementation of any RK method for equation (7) leads to

$$\mathbf{y}_{k+1} = K(\mathbf{z})\mathbf{y}_k, \tag{8}$$

where

$$\begin{aligned}
&\mathbf{z} = diag(z_1, z_2, \ldots, z_m), \\
&z_i = \Delta t q(t_n + \eta_i \Delta t), \quad i = 1, 2, \ldots, m, \\
&K(\mathbf{z}) = 1 + \theta^T \mathbf{z}(I - \Lambda\mathbf{z})^{-1}\mathbb{I}, \quad \mathbb{I} = (1, 1, \ldots, 1)^T, \quad i = 1, 2, \ldots, m.
\end{aligned}$$

Definition 1. *[24] A numerical method is said to be AN-stable if*

$$| K(\mathbf{z}) |< 1, \quad \forall \Re(z_i) \leq 0,\ i = l,2,\dots,m, \tag{9}$$

and A-stable if $q(t)$ is constant and

$$| K(z\mathbf{I}) |< 1, \quad z = q\Delta t \leq 0. \tag{10}$$

Definition 2. *[24] An RK method is said to be LN-stable if it is AN-stable and*

$$|K(\mathbf{z})| \xrightarrow[|\mathbf{z}|\to\infty]{} 0. \tag{11}$$

It also becomes L-stable if it is A-stable and

$$|K(z\mathbf{I})| \xrightarrow[|z|\to\infty]{} 0. \tag{12}$$

2.3 Fractional Calculus

Fractional calculus is a branch of mathematics devoted to the study of integral and differential operators of non-integer order. In this section, we give some preliminary results about fractional calculus [6, 10, 15, 22].

Definition 3. *The $\alpha > 0$ order Riemann-Liouville fractional integral for a function $f \in L_1[a,b]$ is denoted by $J_0^{\alpha} f$ and is defined as*

$$\begin{aligned} J_0^{\alpha} f(x) &= \frac{1}{\Gamma(\alpha)} \int_0^x (x-t)^{\alpha-1} f(t)dt, \\ J_0^{0} f(x) &= f(x), \end{aligned}$$

where $\alpha \in R^{>0}$, $a \leqslant x \leqslant b$ and Γ is Gamma function [22].

Definition 4. *The $\alpha > 0$ order Riemann-Liouville fractional derivative of a function f is defined as*

$$D_{*0}^{\alpha} f(x) = D^n J_0^{n-\alpha} f(x) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dx^n} \int_0^x (x-t)^{n-\alpha-1} f(t)dt,$$

in which $n-1 < \alpha \leqslant n$, $n \in \mathbb{N}$ and $x > 0$ [6].

Definition 5. *The $\alpha > 0$ order fractional derivative of a function f in the Caputo sense is defined as*

$$D_0^{\alpha} f(x) = J_0^{n-\alpha} D^n f(x) = \frac{1}{\Gamma(n-\alpha)} \int_0^x (x-t)^{n-\alpha-1} \frac{d^n f(t)}{dt^n} dt, \tag{13}$$

where $n-1 < \alpha \leqslant n$, $n \in \mathbb{N}$ and $x > 0$ [6, 10].

Definition 6. *The partial Riemann-Liouville fractional integral operator of order $\alpha > 0$ with respect to t is defined as*

$$\begin{aligned} J_{0t}^{\alpha} f(x,t) &= \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(x,\tau)d\tau, \quad (x,t) \in [0,a] \times [0,b] \\ J_0^{0} f(x) &= f(x) \end{aligned} \tag{14}$$

for $n-1 < \alpha \leqslant n$, $n \in \mathbb{N}$ and $x > 0$ [15].

Definition 7. *The Riemann-Liouville partial fractional derivative of order* $\alpha > 0$ *is defined by*

$$
\begin{aligned}
D^{\alpha}_{*0t} f(x,t) &= \frac{\partial^{\alpha}}{\partial t^{\alpha}} f(x,t) = \frac{\partial^{n}}{\partial t^{n}} J^{n-\alpha}_{0t} f(x,t) \\
&= \begin{cases} \dfrac{1}{\Gamma(n-\alpha)} \dfrac{\partial^{n}}{\partial t^{n}} \displaystyle\int_0^t (t-\tau)^{n-\alpha-1} f(x,\tau) d\tau, & n-1 < \alpha \leqslant n, \\ \dfrac{\partial^{n} f(x,t)}{\partial t^{n}}, & \alpha = n \in \mathbb{N}, \end{cases}
\end{aligned} \tag{15}
$$

for $n-1 < \alpha \leqslant n, n \in \mathbb{N}$ *and* $x > 0$ *[15].*

Definition 8. *The Liouville-Caputo partial fractional derivative of order* $\alpha > 0$ *is defined by*

$$
\begin{aligned}
D^{\alpha}_{0t} f(x,t) &= \frac{\partial^{\alpha}}{\partial t^{\alpha}} f(x,t) = J^{n-\alpha}_{0t} \frac{\partial^{n}}{\partial t^{n}} f(x,t) \\
&= \begin{cases} \dfrac{1}{\Gamma(n-\alpha)} \displaystyle\int_0^t (t-\tau)^{n-\alpha-1} \dfrac{\partial^{n} f(x,\tau)}{\partial \tau^{n}} d\tau, & n-1 < \alpha \leqslant n, \\ \dfrac{\partial^{n} f(x,t)}{\partial t^{n}}, & \alpha = n \in \mathbb{N}, \end{cases}
\end{aligned} \tag{16}
$$

for $n-1 < \alpha \leqslant n, n \in \mathbb{N}$ *and* $x > 0$ *[15].*

In the following, some important properties of fractional calculations are listed [6]

- $J^{\alpha}_{0} J^{\beta}_{0} f(x) = J^{\beta}_{0} J^{\alpha}_{0} f(x) = J^{\alpha+\beta}_{0} f(x)$,
- $D^{\alpha}_{0r} J^{\alpha}_{0t} f(x,t) = f(x,t)$,
- $\mathrm{J}^{\alpha}_{0} x^{v} = \frac{\Gamma(v+1)}{\Gamma(\alpha+v+1)} x^{\alpha+v}$,

where $f \in L_1[a,b]$, $\alpha, \beta > 0$ and $v > -1$.

3 The proposed method

As mentioned previously, we will extend the MOLs to solve FPIDE (1) - (2):

$$
\begin{aligned}
&D^{\alpha}_{0t} u(x,t) = f(x,t) + \int_0^t \int_0^x k(x,t,y,z) u(y,z) dy dz, \\
&u(x,0) = h_0(x),
\end{aligned}
$$

where $(x,t)\in[0,a]\times[0,b]$ and $0<\alpha<1$. Using the fractional calculations rules and changing the order of integration, we have

$$\begin{aligned}
D_{0t}^{\alpha}u(x,t) =&f(x,t)+D_{0t}^{\alpha}J_{0t}^{\alpha}\int_0^t\int_0^x k(x,t,y,z)u(y,z)dydz,\\
=&f(x,t)+\frac{1}{\Gamma(\alpha)}D_{0t}^{\alpha}\int_0^t(t-s)^{\alpha-1}\int_0^s\int_0^x k(x,s,y,z)u(y,z)dydzds\\
=&f(x,t)+\frac{1}{\Gamma(\alpha)}D_{0t}^{\alpha}\int_0^t\int_0^x\int_z^t(t-s)^{\alpha-1}k(x,s,y,z)u(y,z)dsdydz,\\
=&f(x,t)+D_{0t}^{\alpha}\int_0^t\int_0^x J_{zt}^{\alpha}\Big(k(x,t,y,z)\Big)u(y,z)dydz,
\end{aligned}$$

and setting

$$J_{zt}^{\alpha}\Big(k(x,t,y,z)\Big)=\alpha(x,t)\beta(y,z), \tag{17}$$

and

$$p(x,t)=\int_0^t\int_0^x\beta(y,z)u(y,z)dydz, \tag{18}$$

implies

$$D_{0t}^{\alpha}u(x,t)=f(x,t)+D_{0t}^{\alpha}\Big(\alpha(x,t)p(x,t)\Big),$$

or

$$u(x,t)=J_{0t}^{\alpha}f(x,t)+\alpha(x,t)p(x,t)+u(x,0)-\alpha(x,0)p(x,0).$$

Since $p(x,0)=0$ and $u(x,0)=h_0(x)$, thus

$$u(x,t)=g(x,t)+\alpha(x,t)p(x,t), \tag{19}$$

where

$$g(x,t)=h_0(x)+\frac{1}{\Gamma(\alpha)}\int_0^t(t-s)^{\alpha-1}f(x,s)ds. \tag{20}$$

We also have

$$\frac{\partial}{\partial t}p(x,t)=\int_0^x\beta(y,t)u(y,t)dy, \tag{21}$$

and finally substituting from (19) into (21), yields

$$\frac{\partial}{\partial t}p(x,t)=\int_0^x\beta(y,t)g(y,t)dy+\int_0^x\alpha(y,t)\beta(y,t)p(y,t)dy. \tag{22}$$

Now, by applying the MOLs to the VIDE (22), we convert it to a system of ODEs. To this end, consider the uniformly distributed points $x_i=i\frac{a}{N}=ih$, $i=0,1,\ldots,N$ of $[0,a]$ and $t_j=j\frac{b}{M}=jk$, $j=0,1,\ldots,M$ of $[0,b]$. Setting $x=x_i$ for $i=1,2,\ldots,N$, in the equation (22), yields

$$\frac{\partial}{\partial t}p(x_i,t)=\int_0^{x_i}\beta(y,t)g(y,t)dy+\int_0^{x_i}\alpha(y,t)\beta(y,t)p(y,t)dy. \tag{23}$$

In order to compute the integral term of the above equation, we apply the following rule

$$S_{2n}=\frac{1}{3}\left(T_n+2M_n\right),$$

in which S_{2n}, T_n and M_n refer to Simpson's, trapezoidal and midpoint quadrature rules, respectively, where

$$\begin{aligned} S_{2n} &:= \int_0^{x_{2n}} f(x)dx = \frac{h}{3}\left(f(0)+f(x_{2n})+2\sum_{i=1}^{n-1} f(x_{2i})+4\sum_{i=1}^{n} f(x_{2i-1}) \right), \\ T_n &:= \int_0^{x_n} f(x)dx = h\left(\frac{1}{2}(f(0)+f(x_n))+\sum_{i=1}^{n-1} f(x_i) \right), \\ M_n &:= \int_0^{x_n} f(x)dx = h\sum_{i=0}^{n-1} f(\frac{x_i+x_{i+1}}{2}). \end{aligned}$$

On the other hand, the interpolation polynomial of $f(x)$ at data points $(x_i, f(x_i))$, $i = 0,1,\dots,n$ in the Lagrange form is as following

$$L(x) = \sum_{i=0}^{n} f(x_i) l_{n,i}(x), \quad l_{n,i}(x) = \prod_{\substack{0\le m\le n \\ m\neq i}} \frac{x-x_m}{x_i-x_m}.$$

Now we replace $f(\frac{x_i+x_{i+1}}{2})$ with $L(\frac{x_i+x_{i+1}}{2})$, thus

$$\begin{aligned} \tilde{M}_n :=& h\sum_{i=0}^{n-1} L(\frac{x_i+x_{i+1}}{2}) \\ \tilde{S}_{2n} :=& \frac{1}{3}\left(T_n+2\tilde{M}_n\right) = \frac{2}{3}h\left[\sum_{j=1}^{n-1}\left(\frac{1}{2}+\sum_{k=0}^{n-1} l_{n,j}(\frac{x_k+x_{k+1}}{2})\right) f(x_j) + \left(\frac{1}{4}+\sum_{k=0}^{n-1} l_{n,0}(\frac{x_k+x_{k+1}}{2})\right) f(x_0) \right. \\ & \left. + \left(\frac{1}{4}+\sum_{k=0}^{n-1} l_{n,n}(\frac{x_k+x_{k+1}}{2})\right) f(x_n) \right] = \sum_{j=0}^{N} w_{n,j} f(x_j), \end{aligned}$$

where

$$\begin{aligned} w_{n,j} =& \frac{2}{3}h\left(\frac{1}{2}+\sum_{k=0}^{n-1} l_{n,j}(\frac{x_k+x_{k+1}}{2}) \right), \quad j = 1,2,\dots,n-1 \\ w_{n,j} =& \frac{2}{3}h\left(\frac{1}{4}+\sum_{k=0}^{n-1} l_{n,j}(\frac{x_k+x_{k+1}}{2}) \right), \quad j = 0,n, \\ w_{n,j} =& 0, \quad j = n+1, n+2, \dots, N. \end{aligned}$$

According to the above results, we obtain the following system of ODEs for nodal unknowns $p(x_j,t)$ and rewrite the equation (23) as follow

$$\frac{\partial p}{\partial t}(x_i,t) = \int_0^{x_i} \beta(y,t)g(y,t)dy + \sum_{j=0}^{i} w_{i,j}\alpha(x_j,t)\beta(x_j,t)p(x_j,t), \quad i = 1,2,\dots,N. \tag{24}$$

Since $p(x_0,t) = 0$, we have

$$\frac{\partial p}{\partial t}(x_i,t) = \int_0^{x_i} \beta(y,t)g(y,t)dy + \sum_{j=1}^{i} w_{i,j}\alpha(x_j,t)\beta(x_j,t)p(x_j,t), \quad i = 1,2,\dots,N, \tag{25}$$

which can be written in matrix form as

$$\begin{aligned} &\mathbf{p}'(t) = \mathbf{A}(t)\mathbf{p}(t) + \mathbf{b}(t), \\ &\mathbf{p}(0) = \mathbf{0}, \end{aligned} \tag{26}$$

where

$$\mathbf{A}_{i,j}(t) = \begin{cases} w_{i,j}\alpha(x_j,t)\beta(x_j,t), & j = 1,2,\cdots,i, \\ & \\ 0, & o.w, \end{cases} \qquad i = 1,2,\ldots,N,$$

$$\mathbf{b}(t) = \left(\int_0^{x_1} \beta(y,t)g(y,t)dy, \int_0^{x_2} \beta(y,t)g(y,t)dy, \ldots, \int_0^{x_N} \beta(y,t)g(y,t)dy\right)^T,$$

$$\mathbf{p}(t) = \Big(p(x_1,t), p(x_2,t), \ldots, p(x_N,t)\Big)^T.$$

Now, equation (26) should be solved. It can be solved by an analytical or a numerical method to obtain a semi-analytical or an approximate solution, respectively. These two cases are described in the following two subsections.

The existence and uniqueness of the solution of the ODEs are given in the following theorem.

Theorem 1. *[1] Suppose that the $n \times n$ matrix function $\mathbf{A}(t)$ and the $n \times 1$ matrix function $\mathbf{b}(t)$ are both continuous on an interval I in $\mathbb{R}$ and $t_0 \in I$. Then, for every choice of the vector $\mathbf{p}_0$, the initial value problem*

$$\begin{aligned} &\mathbf{p}'(t) = \mathbf{A}(t)\mathbf{p}(t) + \mathbf{b}(t), \\ &\mathbf{p}(t_0) = \mathbf{p}_0, \end{aligned}$$

has a unique solution $\mathbf{p}(t)$ which is defined on the same interval I.

3.1 Semi-analytical solution

The system (26) can be solved analytically as follows [1]:

$$\mathbf{p}(t) = e^{\int_0^t \mathbf{A}(z)dz}\mathbf{p}_0 + e^{\int_0^t \mathbf{A}(z)dz}\int_0^t e^{-\int_0^s \mathbf{A}(z)dz}\mathbf{b}(s)ds, \tag{27}$$

or

$$\mathbf{p}(t) = e^{\int_0^t \mathbf{A}(z)dz}\mathbf{p}_0 + \int_0^t \mathbf{R}(t,s)\mathbf{b}(s)ds, \tag{28}$$

where $\mathbf{R}(t,s) = e^{\int_s^t \mathbf{A}(z)dz}$ is the resolvent kernel of the system (26). Now, setting $x = x_i$ in (19) yields

$$u(x_i,t) = g(x_i,t) + \alpha(x_i,t)p(x_i,t), \quad i = 1,2,\ldots,N, \tag{29}$$

which is a semi-analytical solution of (19) and in matrix-vector form can be written as

$$\mathbf{u}(t) = \mathbf{g}(t) + \mathbf{B}(t)\mathbf{p}(t), \tag{30}$$

where

$$\mathbf{u}(t) = \Big(u(x_1,t), u(x_2,t), \ldots, u(x_N,t)\Big)^T, \tag{31}$$

$$\mathbf{g}(t) = \Big(g(x_1,t), g(x_2,t), \ldots, g(x_N,t)\Big)^T, \tag{32}$$

$$\mathbf{B}(t) = \operatorname{diag}\Big(\alpha(x_1,t), \alpha(x_2,t), \ldots, \alpha(x_N,t)\Big). \tag{33}$$

3.2 Numerical Solution

In this section, two classes of numerical methods, including Radau IIA and Lobatto IIIC of implicit RK methods will be discussed to solve the system of ODEs (26). If we apply an m-stage RK method to solve the initial value problem (26); we have the formulae with step-size k in the interval $[0,b]$ as

$$\begin{aligned}
\mathbf{R}_{1,n} &= \mathbf{p}_n + k\sum_{j=1}^{m} \lambda_{i,j}\Big(\mathbf{A}(t_n+\eta_j k)\mathbf{R}_{j,n} + \mathbf{b}(t_n+\eta_j k)\Big),\\
&\vdots\\
\mathbf{R}_{m,n} &= \mathbf{p}_n + k\sum_{j=1}^{m} \lambda_{i,j}\Big(\mathbf{A}(t_n+\eta_j k)\mathbf{R}_{j,n} + \mathbf{b}(t_n+\eta_j k)\Big),\\
\mathbf{p}_{n+1} &= \mathbf{p}_n + k\sum_{j=1}^{m} \theta_j\Big(\mathbf{A}(t_n+\eta_j k)\mathbf{R}_{j,n} + \mathbf{b}(t_n+\eta_j k)\Big).
\end{aligned} \tag{34}$$

3.2.1 Radau IIA and Lobatto IIIC Runge-Kutta Methods

The m-stage Radau IIA and Lobatto IIIC methods are two classes of RK methods of non-stiff orders $2m-1$ and $2m-2$, respectively. They are L-stable and excellent methods for stiff problems. Butcher tables of these methods are given in Tables 2 and 3 [13].

Table 2: Radau IIA methods with $m = 2$ and $m = 3$.

$\frac{1}{3}$	$\frac{5}{12}$	$-\frac{1}{12}$
1	$\frac{3}{4}$	$\frac{1}{4}$
	$\frac{3}{4}$	$\frac{1}{4}$

Radau IIA method ($m = 2$)

$\frac{4-\sqrt{6}}{10}$	$\frac{88-7\sqrt{6}}{360}$	$\frac{296-169\sqrt{6}}{1800}$	$\frac{-2+3\sqrt{6}}{225}$
$\frac{4+\sqrt{6}}{10}$	$\frac{296+169\sqrt{6}}{1800}$	$\frac{88+7\sqrt{6}}{360}$	$\frac{-2-3\sqrt{6}}{225}$
1	$\frac{16-\sqrt{6}}{36}$	$\frac{16+\sqrt{6}}{36}$	$\frac{1}{9}$
	$\frac{16-\sqrt{6}}{36}$	$\frac{16+\sqrt{6}}{36}$	$\frac{1}{9}$

Radau IIA method ($m = 3$)

In the following, we examine the LN-stability of Radau IIA and Lobatto IIIC RK methods.

Table 3: Lobatto IIIC method with $m = 2$, $m = 3$ and $m = 4$.

$$\begin{array}{c|cc} 0 & \frac{1}{2} & -\frac{1}{2} \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Lobatto IIIC method ($m = 2$)

$$\begin{array}{c|ccc} 0 & \frac{1}{6} & -\frac{1}{3} & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{6} & \frac{5}{12} & -\frac{1}{12} \\ 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

Lobatto IIIC method ($m = 3$)

$$\begin{array}{c|cccc} 0 & \frac{1}{12} & -\frac{\sqrt{5}}{12} & \frac{\sqrt{5}}{12} & -\frac{1}{12} \\ \frac{1}{2}-\frac{\sqrt{5}}{10} & \frac{1}{12} & \frac{1}{4} & \frac{10-7\sqrt{5}}{60} & \frac{\sqrt{5}}{60} \\ \frac{1}{2}+\frac{\sqrt{5}}{10} & \frac{1}{12} & \frac{10+7\sqrt{5}}{60} & \frac{1}{4} & -\frac{\sqrt{5}}{60} \\ 1 & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \\ \hline & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \end{array}$$

Lobatto IIIC method ($m = 4$)

Definition 9. *[4] Consider the matrix* $\mathbf{M}$ *with entries* $\mathbf{M}_{i,j} = \theta_i\lambda_{i,j} + \theta_j\lambda_{j,i} - \theta_i\theta_j$. *An RK method is strongly algebraically stable if* $\mathbf{M}$ *is a non-negative definite matrix,* $\theta_i > 0, i = 1,2,\ldots,m$ *and*

$$0 < \theta^T\Lambda^{-1}\mathbb{I} < 2, \quad \mathbb{I} = (1,\ldots,1)^T.$$

Lemma 1. *[4] If in Definition 9,* $\theta^T\Lambda^{-1}\mathbb{I} = 1$ *the stability property is a generalization of L-stability to the non-autonomous equation.*

Theorem 2. *The Radau IIA and Lobatto IIIC RK methods are LN-stable.*

Proof. For both of the above methods $\mathbf{M}_{i,j} = \theta_i\lambda_{i,j} + \theta_j\lambda_{j,i} - \theta_i\theta_j$, where θ_i and $\lambda_{i,j}$ for $i,j = 1,\ldots,m$ are presented in Tables 3 and 2. The non-negativity definite property is easily achieved, and it can also be shown that the following relation

$$\theta^T\Lambda^{-1}\mathbb{I} = 1, \quad \mathbb{I} = (1,\ldots,1)^T,$$

is established for both methods. According to Definition 9 and Lemma 1 in general, an m-stage RK method is LN-stable if θ_i, $i = 1,2,\ldots,m$ are positive, $\mathbf{M}$ is a non-negative definite matrix, and $\theta^T\Lambda^{-1}\mathbb{I} = 1$. All the above three conditions are satisfied for Radau IIA and Lobatto IIIC Rk methods, therefore these methods are LN-stable. □

4 Stability and Local Truncation Error

In this section, we investigate the stability of the proposed method. The truncation error caused by the proposed method is also analyzed.

4.1 Stability

Stability is one of the most important issues in the numerical solutions of differential equations. To discuss the stability test, let us first introduce the following test equation :

$$\begin{aligned} &D_{0t}^{\alpha}u(x,t)+\frac{1}{\Gamma(1-\alpha)}\int_0^t\int_0^x (t-z)^{-\alpha}b'(y)a'(z)u(y,z)dydz = f(x,t), \\ &u(x,0)=\exp(b(x)i-a(t))\big|_{t=0}, \end{aligned} \tag{35}$$

where $0<\alpha<1$ and

$$f(x,t)=\Big((1+i)\exp\Big(b(x)i\Big)-i\Big)\sum_{l=1}^{\infty}\exp\Big(a(t)\Big)^{(l)}\Big|_{t=0}\frac{t^{l-\alpha}}{\Gamma(l-\alpha+1)}$$

with the exact solution $u(x,t)=\exp\Big(b(x)i-a(t)\Big)$, in which $b(x)>0$, $a'(t)>0$ and $a(t)>0$ is an increasing function. Obviously, $u(x,t)$ tends to zero as t tends to ∞, and we expect that the numerical solution has the same behavior. To solve the above problem, FPIDE (35) should be converted to a VIE as follows

$$u(x,t)=g(x,t)-\int_0^t\int_0^x K(x,t,y,z)u(y,z)dydz,$$

with $(x,t)\in[0,a]\times[0,b]$ and

$$g(x,t)=(i+1)\exp\Big(b(x)i-a(t)\Big)-i\exp\Big(b(x)i-a(0)\Big)-i\exp\Big(b(0)i\Big)\Bigg(\exp\Big(-a(t)\Big)-\exp\Big(-a(0)\Big)\Bigg).$$

According to the existing rules for fractional and classical integrations, if $0<\alpha<1$, we have

$$\mathfrak{I}=\int_0^t\int_0^x K(x,t,y,z)u(y,z)dydz=\frac{1}{\Gamma(1-\alpha)\Gamma(\alpha)}\int_0^t(t-s)^{\alpha-1}\int_0^s\int_0^x(s-z)^{-\alpha}b'(y)a'(z)u(y,z)dydzds.$$

Changing the integral order and substitution $s=z+r(t-z)$ yields

$$\begin{aligned}\mathfrak{I}&=\frac{1}{\Gamma(1-\alpha)\Gamma(\alpha)}\int_0^t\int_0^x\int_z^t(t-s)^{\alpha-1}(s-z)^{-\alpha}b'(y)a'(z)u(y,z)dsdydz\\ &=\frac{1}{\Gamma(1-\alpha)\Gamma(\alpha)}\int_0^t\int_0^x\int_0^1(1-r)^{\alpha-1}r^{-\alpha}b'(y)a'(z)u(y,z)drdydz\\ &=\frac{B(1-\alpha,\alpha)}{\Gamma(1-\alpha)\Gamma(\alpha)}\int_0^t\int_0^x b'(y)a'(z)u(y,z)dydz=\int_0^t\int_0^x b'(y)a'(z)u(y,z)dydz.\end{aligned}$$

Let us consider uniform partitions with step-sizes h and k for the intervals $[0,a]$ and $[0,b]$, respectively. Applying the proposed method to the test equation yields

$$\mathbf{u}(t)=\mathbf{g}(t)+\mathbf{B}(t)\mathbf{p}(t), \tag{36}$$

and

$$\begin{aligned}&\mathbf{p}'(t) = \mathbf{A}(t)\mathbf{p}(t) + \mathbf{b}(t),\\ &\mathbf{p}(0) = \mathbf{0},\end{aligned} \tag{37}$$

where $\mathbf{u}(t)$ and $\mathbf{g}(t)$ were defined in (31) and (33), respectively . Also

$$\mathbf{A}_{i,j}(t) = \begin{cases} -w_{i,j}b'(x_j)a'(t), & j = 1,2,\ldots,i, \\ \\ 0, & o.w, \end{cases} \quad i = 1,2,\ldots,N, \tag{38}$$

and

$$\begin{aligned}\mathbf{b}(t) =& \Big(-\int_0^{x_1} b'(y)a'(t)g(y,t)dy, -\int_0^{x_2} b'(y)a'(t)g(y,t)dy, \ldots, -\int_0^{x_N} b'(y)a'(t)g(y,t) \Big)^T,\\ \mathbf{p}(t) =& \Big(p(x_1,t), p(x_2,t), \ldots, p(x_N,t) \Big)^T,\\ \mathbf{B}(t) =& \text{diag}\Big[\alpha(x_1,t), \alpha(x_1,t), \ldots, \alpha(x_N,t) \Big].\end{aligned}$$

In the sequel, a Radau IIA or Lobatto IIIC RK method with m-stage is used to solve the equation (37) as follows

$$\begin{aligned}\mathbf{R}_{i,n} &= \mathbf{p}_n + k\sum_{j=1}^{m} \lambda_{i,j}\left(\mathbf{A}(t_n+\eta_j k)\mathbf{R}_{j,n} + \mathbf{b}(t_n+\eta_j k)\right), \quad i = l,2,\ldots,m,\\ \mathbf{p}_{n+1} &= \mathbf{p}_n + k\sum_{j=1}^{m} \theta_j\left(\mathbf{A}(t_n+\eta_j k)\mathbf{R}_{j,n} + \mathbf{b}(t_n+\eta_j k)\right).\end{aligned} \tag{39}$$

Consider the equation (39)

$$\begin{aligned}\mathbf{R}_{i,n} - k\sum_{j=1}^{m} \lambda_{i,j}\mathbf{A}(t_n+\eta_j k)\mathbf{R}_{j,n} &= \mathbf{p}_n + k\sum_{j=1}^{m} \lambda_{i,j}\mathbf{b}(t_n+\eta_j k), \quad i = l,2,\ldots,m,\\ \mathbf{p}_{n+1} - k\sum_{j=1}^{m} \theta_j\mathbf{A}(t_n+\eta_j k)\mathbf{R}_{j,n} &= \mathbf{p}_n + k\sum_{j=1}^{m} \theta_j\mathbf{b}(t_n+\eta_j k),\end{aligned} \tag{40}$$

which can be written in matrix form as

$$\begin{pmatrix} \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix} - \begin{pmatrix} k\lambda_{1,1}\mathbf{A}(t_n+\eta_1 k) & k\lambda_{1,2}\mathbf{A}(t_n+\eta_2 k) & \cdots & k\lambda_{1,m}\mathbf{A}(t_n+\eta_m k) \\ k\lambda_{2,1}\mathbf{A}(t_n+\eta_1 k) & k\lambda_{2,2}\mathbf{A}(t_n+\eta_2 k) & \cdots & k\lambda_{2,m}\mathbf{A}(t_n+\eta_m k) \\ \vdots & \vdots & \ddots & \vdots \\ k\lambda_{m,1}\mathbf{A}(t_n+\eta_1 k) & k\lambda_{m,2}\mathbf{A}(t_n+\eta_2 k) & \cdots & k\lambda_{m,m}\mathbf{A}(t_n+\eta_{m-1} k) \end{pmatrix} & 0 \\ \begin{pmatrix} \theta_1 & \theta_2 & \cdots & \theta_m \end{pmatrix} \begin{pmatrix} -k\mathbf{A}(t_n+\eta_1 k) & & & \\ & -k\mathbf{A}(t_n+\eta_2 k) & & \\ & & \ddots & \\ & & & -k\mathbf{A}(t_n+\eta_m k) \end{pmatrix} & 1 \end{pmatrix}$$

$$\times \begin{pmatrix} \begin{pmatrix} \mathbf{R}_{1,n} \\ \mathbf{R}_{2,n} \\ \vdots \\ \mathbf{R}_{m,n} \end{pmatrix} \\ \mathbf{p}_{n+1} \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \\ 1 \end{pmatrix} \mathbf{p}_n + \begin{pmatrix} \begin{pmatrix} \lambda_{1,1} & \lambda_{1,2} & \cdots & \lambda_{1,m} \\ \lambda_{2,1} & \lambda_{2,2} & \cdots & \lambda_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{m,1} & \lambda_{m,2} & \cdots & \lambda_{m,m} \end{pmatrix} \begin{pmatrix} k\mathbf{b}(t_n+\eta_1 k) \\ k\mathbf{b}(t_n+\eta_2 k) \\ \ddots \\ k\mathbf{b}(t_n+\eta_m k) \end{pmatrix} \\ \begin{pmatrix} \theta_1 & \theta_2 & \cdots & \theta_m \end{pmatrix} \begin{pmatrix} k\mathbf{b}(t_n+\eta_1 k) \\ k\mathbf{b}(t_n+\eta_2 k) \\ \ddots \\ k\mathbf{b}(t_n+\eta_m k) \end{pmatrix} \end{pmatrix}.$$

Assume that

$$\mathbf{Z} = diag(\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_m), \quad \mathbf{Z}_i = k\mathbf{A}(t_n+\eta_i k), \quad i = 1, 2, \ldots, m,$$

$$\Psi = \begin{pmatrix} k\mathbf{b}(t_n+\eta_1 k) \\ k\mathbf{b}(t_n+\eta_2 k) \\ \vdots \\ k\mathbf{b}(t_n+\eta_m k) \end{pmatrix}, \quad \Phi = \begin{pmatrix} \mathbf{R}_{1,n} \\ \mathbf{R}_{2,n} \\ \vdots \\ \mathbf{R}_{m,n} \end{pmatrix}.$$

Therefore

$$\begin{pmatrix} (\mathbf{I} - \Lambda\mathbf{Z}) & 0 \\ -\theta^T\mathbf{Z} & 1 \end{pmatrix} \begin{pmatrix} \Phi \\ \mathbf{p}_{n+1} \end{pmatrix} = \begin{pmatrix} \mathbb{I} \\ 1 \end{pmatrix} \mathbf{p}_n + \begin{pmatrix} \Lambda\Psi \\ \theta^T\Psi \end{pmatrix},$$

Eliminating the internal stages Φ_i in (39) results in

$$\mathbf{p}_{n+1} = \mathbf{K}(\mathbf{Z})\mathbf{p}_n + \mathbf{S}(\mathbf{Z})\Psi, \tag{41}$$

$$\mathbf{K}(\mathbf{Z}) = 1 + \theta^T\mathbf{Z}(\mathbf{I} - \Lambda\mathbf{Z})^{-1}\mathbb{I}, \tag{42}$$

$$\mathbf{S}(\mathbf{Z}) = (\mathbf{q}_1(\mathbf{Z}), \mathbf{q}_2(\mathbf{Z}), \ldots, \mathbf{q}_m(\mathbf{Z})) = \theta^T\mathbf{Z}(\mathbf{I} - \Lambda\mathbf{Z})^{-1}\Lambda + \theta^T. \tag{43}$$

On the other hand, by using (30) and (41)-(43), we can write

$$\begin{aligned} \mathbf{u}_{n+1} &= \mathbf{g}(t_{n+1}) + \mathbf{p}_{n+1}, \\ \mathbf{u}_{n+1} &= \mathbf{g}(t_{n+1}) + \mathbf{K}(\mathbf{Z})\mathbf{p}_n + \mathbf{S}(\mathbf{Z})\Psi, \\ &= \mathbf{g}(t_{n+1}) + \mathbf{K}(\mathbf{Z})\,(\mathbf{u}_n - \mathbf{g}(t_n)) + \mathbf{S}(\mathbf{Z})\Psi, \\ &= \mathbf{K}(\mathbf{Z})\mathbf{u}_n - \mathbf{K}(\mathbf{Z})\mathbf{g}(t_n) + \mathbf{S}(\mathbf{Z})\Psi + \mathbf{g}(t_{n+1}). \end{aligned} \tag{44}$$

By substituting $\mathbf{p}_0$ with $\mathbf{p}_0^*$ and by using any RK method to solve the system (37), we have

$$\begin{aligned}
&\mathbf{p^*}_{n+1} = \mathbf{K}(\mathbf{Z})\mathbf{p^*}_n + \mathbf{S}(\mathbf{Z})\Psi,\\
&\mathbf{u^*}_{n+1} = \mathbf{g}(t_{n+1}) + \mathbf{p^*}_{n+1},\\
&\mathbf{u}^*_{n+1} = \mathbf{K}(\mathbf{Z})\mathbf{u}^*_n - \mathbf{K}(\mathbf{Z})\mathbf{g}(t_n) + \mathbf{S}(\mathbf{Z})\Psi + \mathbf{g}(t_{n+1}).
\end{aligned} \tag{45}$$

Now by subtracting (45) from (44) and by setting $E_{\mathbf{p}}(t_{n+1}) = \mathbf{p}^*_{n+1} - \mathbf{p}_{n+1}$ and $E_{\mathbf{u}}(t_{n+1}) = \mathbf{u}^*_{n+1} - \mathbf{u}_{n+1}$, we have

$$E_{\mathbf{u}}(t_{n+1}) = \mathbf{K}(\mathbf{Z})E_{\mathbf{u}}(t_n). \tag{46}$$

So, according to Definition 2 and Theorem 2, if we apply LN-stable RK method to solve the non-autonomous equation (37), the error of numerical solution VIE satisf ies in equation (46), and the proposed method is LN-stable for VIE. With this argument, every FPIDE is equivalent to a VIE, so it can be inferred that the presented method is LN-stable for FPIDE.

4.2 Local Truncation Error

Local truncation error analysis in a numerical method is one of the most important information about the theoretical specifications that distinguish more efficient numerical methods from the others. It is the amount of error that occurs in a single step of a numerical approximation. It can be evaluated as the difference between the approximate and exact solutions in one step of a numerical method. Therefore local truncation error τ_{n+1} is

$$\tau_n = u(t_n) - u_n = O(\Delta t^{p+1}),$$

where $u(t_n)$ is the exact solution at time t_n, u_n is the approximate solution in step n and p is order of the numerical method. In this section, local truncation errors of considered numerical schemes are studied to solve ODE (26). The accuracy and local truncation error of Lobatto IIIC RK method with m-stage for stiff and non-stiff ODEs based on the time-step size (k) are $O(k^{m-1})$ and $O(k^{2m-2})$, respectively [17]. Similarly, the local truncation errors of the Radau IIA RK method are $O(k^m)$ and $O(k^{2m-1})$, for stiff and non-stiff ODEs, respectively [17]. We know

$$\begin{aligned}
&\mathbf{p}'(t) = \mathbf{A}(t)\mathbf{p}(t) + \mathbf{b}(t) + O(h^3)\mathbb{I},\\
&\mathbf{p}(t_n) = \mathbf{p}_n + O(k^{m-1})\mathbb{I} + O(h^3)\mathbb{I}.
\end{aligned}$$

So local truncation error of obtained numerical solutions by using the Lobatto IIIC RK method for the FPIDE (1) and (2) in the stiff case is as following

$$\begin{aligned}
\mathbf{u}(t_n) =& \mathbf{g}(t_n) + \mathbf{B}(t_n)\mathbf{p}(t_n) = \mathbf{g}(t_n) + \mathbf{B}(t_n)\left[\mathbf{p}_n + \Big(O(k^{m-1}) + O(h^3)\Big)\mathbb{I}\right],\\
\mathbf{u}_n =& \mathbf{g}(t_n) + \mathbf{B}(t_n)\mathbf{p}_n.
\end{aligned}$$

Therefore local truncation error vector(τ_n) in the stiff case is

$$\tau_n = \mathbf{u}(t_n) - \mathbf{u}_n = \Big(O(k^{m-1}) + O(h^3)\Big)\mathbb{I},$$

and in the non-stiff case is $\tau_n = \mathbf{u}(t_n) - \mathbf{u}_n = \Big(O(k^{2m-2}) + O(h^3)\Big)\mathbb{I}$. Similarly, the local truncation errors of Radau IIA RK method for the stiff case is $\tau_n = \mathbf{u}(t_n) - \mathbf{u}_n = \Big(O(k^m) + O(h^3)\Big)\mathbb{I}$, and in the non-stiff case is $\tau_n = \mathbf{u}(t_n) - \mathbf{u}_n = \Big(O(k^{2m-1}) + O(h^3)\Big)\mathbb{I}$.

5 Numerical examples

In this section, several numerical examples are solved by the proposed method and compared with the results obtained in [15]. The comparison of numerical results show high accuracy of our results with respect to the results of [15]. All computations are performed in Maple software 2018.

Example 1. Consider the following TFPIDE with the exact solution $u(x,t) = \exp(-t^2 - t + (x+1/3)^3 i)$ (i is imaginary part)

$$D_{0t}^{1/2}u(x,t) = f(x,t) - \frac{1/\Gamma(1/2)}{120}\int_0^t\int_0^x \frac{(2z+1)(y+1/3)^2}{\sqrt{t-z}}u(y,z)dydz,$$

where $(x,t) \in [0,1]\times[0,1]$ and

$$f(x,t) = \left((1+\frac{i}{3\Gamma(6)})\exp\Big((x+\frac{1}{3})^3 i\Big) - \frac{i}{3\Gamma(6)}\right)\sum_{l=1}^{\infty}\exp(-t^2-t)^{(l)}\Big|_{t=0}\frac{t^{l-1/2}}{\Gamma(l+1/2)}.$$

This equation was solved by the method of [15], semi-analytical and numerical methods (Lobatto IIIC RK method with $m = 2,3,4$ and Radau IIA RK method with $m = 2,3$), described in this paper. The numerical results, including absolute errors in some arbitrary points and maximum value of errors are reported in Tables 4, 5, 6 and 7 . Also, absolute errors , exact and approximate solutions at $(x,t) \in [0,1]\times[0,1]$ are ploted in Figures 1, 2, 3 and 4.

Table 4: Absolute errors in some points for Example 1 by the semi-analytical method and the method of [15].

	semi-analytical method	method of [15]
(x,t)	$N = 10$	$N = 5$
$(0.3,0.5)$	$0.1e-9$	$0.1e-2$
$(0.2,1)$	$0.1e-9$	$0.3e-3$
$(0.9,0.5)$	$0.1e-10$	$0.9e-3$
$(1,0.7)$	$0.2e-9$	$0.8e-2$
$(0.3,0.6)$	$0.2e-10$	$0.1e-2$
$(0.2,0.9)$	$0.1e-9$	$0.1e-3$
$(0.7,0.1)$	$0.5e-12$	$0.2e-2$

Example 2. As second example, consider the following TFPIDE with the exact solution $u(x,t) = \sin(x)\exp(-t^2)$

$$D_{0t}^{2/3}u(x,t) = f(x,t) - \frac{1/\Gamma(16/3)}{\beta(1/3,5)}\int_0^t\int_0^x \frac{\sin(y)z}{\sqrt[3]{(t-z)^2}}u(y,z)dydz,$$

Table 5: Absolute errors in some points for Example 1 by Lobatto IIIC method with $m=2,3,4$.

	Lobatto IIIC method with $m=2$		Lobatto IIIC method with $m=3$		Lobatto IIIC method with $m=4$	
(x,t)	$N=10$ $M=10$	$N=10$ $M=100$	$N=10$ $M=10$	$N=10$ $M=100$	$N=10$ $M=10$	$N=10$ $M=100$
$(0.3,0.5)$	$0.9e-6$	$0.1e-7$	$0.3e-8$	$0.5e-9$	$0.4e-8$	$0.5e-9$
$(0.2,1)$	$0.5e-6$	$0.5e-8$	$0.2e-8$	$0.3e-9$	$0.2e-8$	$0.3e-9$
$(0.9,0.5)$	$0.7e-5$	$0.1e-6$	$0.3e-6$	$0.4e-7$	$0.3e-6$	$0.4e-7$
$(1,0.7)$	$0.9e-5$	$0.1e-6$	$0.9e-6$	$0.9e-7$	$0.8e-6$	$0.9e-7$
$(0.3,0.6)$	$0.1e-5$	$0.1e-7$	$0.5e-8$	$0.6e-9$	$0.5e-8$	$0.6e-9$
$(0.2,0.9)$	$.5e-6$	$.5e-8$	$0.1e-8$	$0.3e-9$	$0.1e-8$	$0.2e-9$
$(0.7,0.1)$	$0.1e-5$	$0.1e-7$	$0.1e-7$	$0.2e-8$	$0.1e-7$	$0.1e-8$

Table 6: Absolute errors in some points for Example 1 by Radau IIA method with $m=2,3$.

	Radau IIA method with $m=2$		Radau IIA method with $m=3$	
(x,t)	$N=10$ $M=10$	$N=10$ $M=100$	$N=10$ $M=10$	$N=10$ $M=100$
$(0.3,0.5)$	$0.4e-8$	$0.5e-9$	$0.4e-8$	$0.5e-9$
$(0.2,1)$	$0.7e-8$	$0.3e-9$	$0.2e-9$	$0.3e-9$
$(0.9,0.5)$	$0.3e-6$	$0.4e-7$	$0.3e-6$	$0.4e-7$
$(1,0.7)$	$0.8e-6$	$0.9e-7$	$0.9e-6$	$0.9e-7$
$(0.3,0.6)$	$0.6e-8$	$0.6e-9$	$0.5e-8$	$0.6e-9$
$(0.2,0.9)$	$0.6e-8$	$0.3e-9$	$0.1e-8$	$0.1e-8$
$(0.7,0.1)$	$0.1e-7$	$0.1e-8$	$0.1e-7$	$0.2e-8$

Table 7: Maximum value of errors for Example 1 by Radau IIA method with $m=2,3$ and Lobatto IIIC method with $m=2,3,4$.

	Radau IIA method		Lobatto IIIC method		
(N,M)	$m=2$	$m=3$	$m=2$	$m=3$	$m=4$
$(10,10)$	$0.1e-5$	$0.1e-5$	$0.9e-5$	$0.1e-5$	$0.1e-5$
$(10,100)$	$0.1e-6$	$0.1e-6$	$0.2e-6$	$0.1e-6$	$0.1e-6$

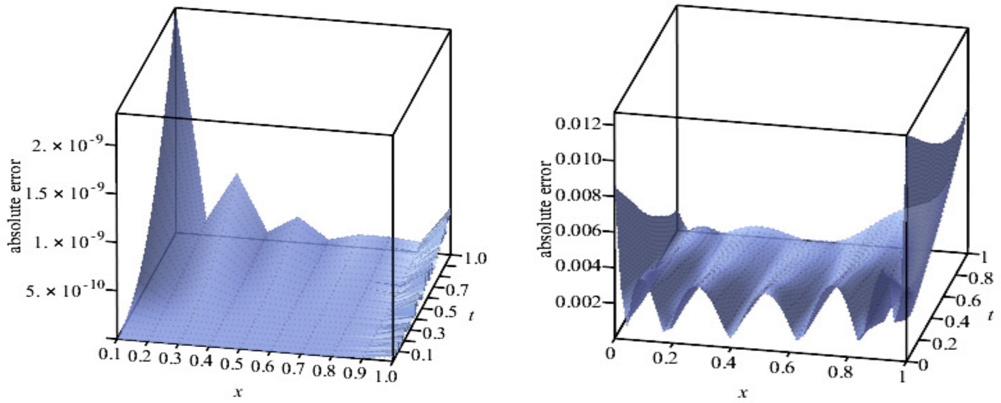

Figure 1: Comparison of absolute error , between the semi-analytical method with $N = 10$ (left) and the method of [15] with $N = 5$ (right) for Example 1.

where $(x,t) \in [0,1] \times [0,1]$ and

$$f(x,t) = \left(\sin(x) - \frac{1/\Gamma(16/3)}{2\Gamma(5)}\sin(x)\right)\sum_{l=1}^{\infty} \exp(-t^2)^{(l)}\Big|_{t=0} \frac{t^{l-2/3}}{\Gamma(l+1/3)}.$$

We proceed as in the previous example and report numerical results in Tables 8, 9, 10 and 11 and Figures 5, 6, 7 and 8.

Table 8: Absolute errors in some points for Example 2 by the semi- analytical method and the methods of [15].

	semi-analytical method	method of [15]
(x,t)	$N = 10$	$N = 5$
$(0.3,0.5)$	$0.4e-10$	$0.5e-5$
$(0.2,1)$	$0.1e-9$	$0.5e-4$
$(0.9,0.5)$	$0.3e-8$	$0.4e-4$
$(1,0.7)$	$0.1e-7$	$0.2e-3$
$(0.3,0.6)$	$0.8e-10$	$0.9e-5$
$(0.2,0.9)$	$0.1e-9$	$0.3e-4$
$(0.7,0.1)$	$0.4e-11$	$0.4e-5$

6 Conclusion

In this article, the method of lines is extended to convert time FPIDE with initial conditions to a system of ODEs. Then LN-stable schemes of the Runge-Kutta method is extended to solve the resulted ODEs system. It is proved that the proposed method is LN-stable, too. It seems that the proposed method can be applied to solve Fredholm and Volterra integral equations of the first kind.

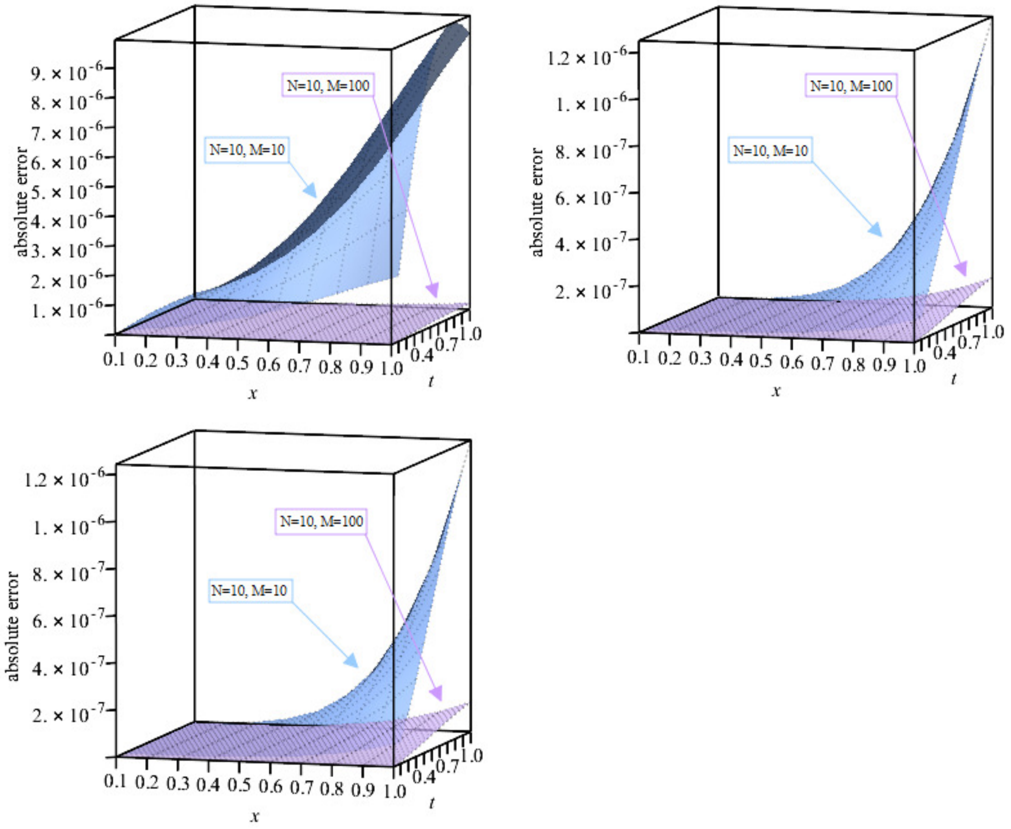

Figure 2: Comparison of absolute error , between $N = 10, M = 10$ and $N = 10, M = 100$ by the Lobatto IIIC method with $m = 2$ (row:1, column:1), $m = 3$ (row:1, column:2) and $m = 4$ (row:2, column:1) for Example 1.

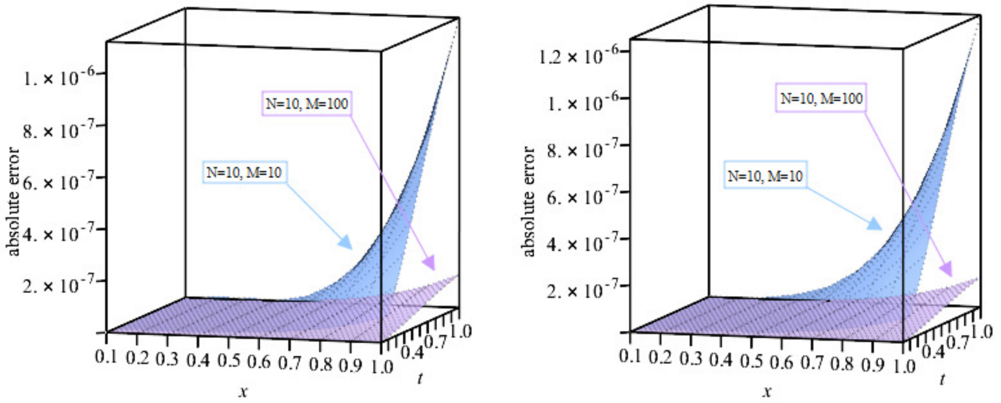

Figure 3: Comparison of absolute error , between $N = 10, M = 10$ and $N = 10, M = 100$ by the Lobatto IIIC method with $m = 2$ (left) and $m = 3$ (right) for Example 1.

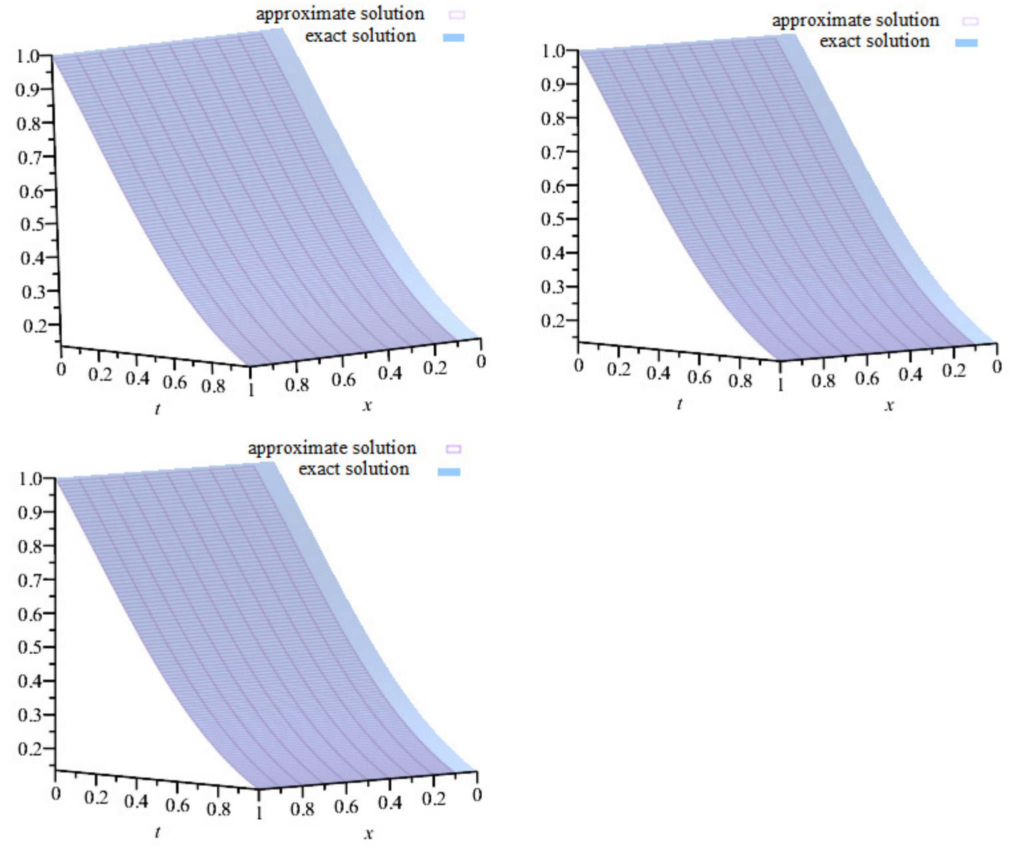

Figure 4: Comparison of exact and approximate solutions by the semi-analytical method with $N = 10$ (row:1, column:1), Lobatto IIIC method with $m = 4, N = 10, M = 100$ (row:1, column:2) and Radau IIA method with $m = 3, N = 10, M = 100$ (row:2, column:1) for Example 1.

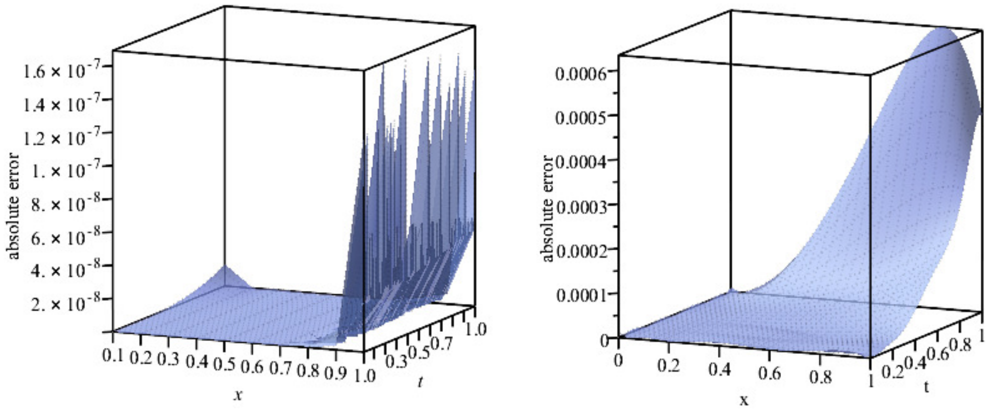

Figure 5: Comparison of absolute error , between the semi-analytical method with $N = 10$ (left) and the method of [15] with $N = 5$ (right) for Example 2.

Table 9: Absolute errors in some points for Example 2 by Lobatto IIIC method with $m = 2,3,4$.

	Lobatto IIIC method with $m = 2$		Lobatto IIIC method with $m = 3$		Lobatto IIIC method with $m = 4$	
(x,t)	$N = 10$ $M = 10$	$N = 10$ $M = 100$	$N = 10$ $M = 10$	$N = 10$ $M = 100$	$N = 10$ $M = 10$	$N = 10$ $M = 100$
$(0.3,0.5)$	$0.9e-6$	$0.1e-7$	$0.7e-8$	$0.8e-9$	$0.8e-8$	$0.7e-9$
$(0.2,1)$	$0.9e-6$	$0.1e-7$	$0.1e-7$	$0.9e-9$	$0.9e-8$	$0.8e-9$
$(0.9,0.5)$	$0.6e-5$	$0.8e-7$	$0.2e-6$	$0.2e-7$	$0.2e-6$	$0.2e-7$
$(1,0.7)$	$0.1e-4$	$0.1e-6$	$0.6e-6$	$0.6e-7$	$0.6e-6$	$0.6e-7$
$(0.3,0.6)$	$0.1e-5$	$0.1e-7$	$0.1e-7$	$0.1e-8$	$0.1e-7$	$0.1e-8$
$(0.2,0.9)$	$0.8e-6$	$0.9e-8$	$0.8e-8$	$0.7e-9$	$0.7e-8$	$0.6e-9$
$(0.7,0.1)$	$0.2e-6$	$0.2e-8$	$0.7e-9$	$0.1e-9$	$0.8e-9$	$0.1e-9$

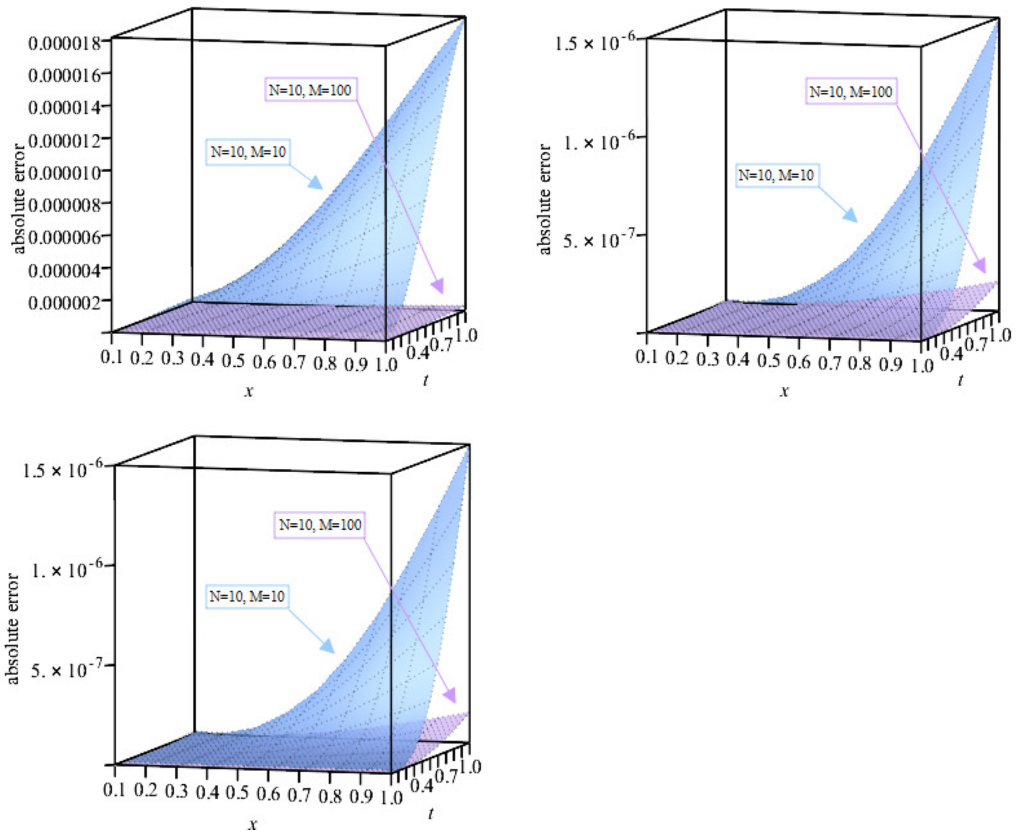

Figure 6: Comparison of absolute error, between $N = 10, M = 10$ and $N = 10, M = 100$ by Lobatto IIIC method with $m = 2$ (row:1, column:1), $m = 3$ (row:1, column:2) and $m = 4$ (row:2, column:1) for Example 2.

Table 10: Absolute errors in some points for Example 2 by Radau IIA method with $m = 2,3$.

	Radau IIA method with m =2		Radau IIA method with m =3	
(x,t)	$N = 10$ $M = 10$	$N = 10$ $M = 100$	$N = 10$ $M = 10$	$N = 10$ $M = 100$
$(0.3,0.5)$	$0.2e-7$	$0.8e-9$	$0.8e-8$	$0.8e-9$
$(0.2,1)$	$0.1e-7$	$0.9e-9$	$0.3e-7$	$0.9e-9$
$(0.9,0.5)$	$0.3e-6$	$0.2e-7$	$0.2e-6$	$0.2e-7$
$(1,0.7)$	$0.8e-6$	$0.7e-7$	$0.6e-6$	$0.6e-7$
$(0.3,0.6)$	$0.2e-7$	$0.1e-8$	$0.1e-7$	$0.1e-8$
$(0.2,0.9)$	$0.1e-7$	$0.7e-9$	$0.8e-8$	$0.7e-9$
$(0.7,0.1)$	$0.2e-7$	$0.1e-9$	$0.8e-9$	$0.1e-9$

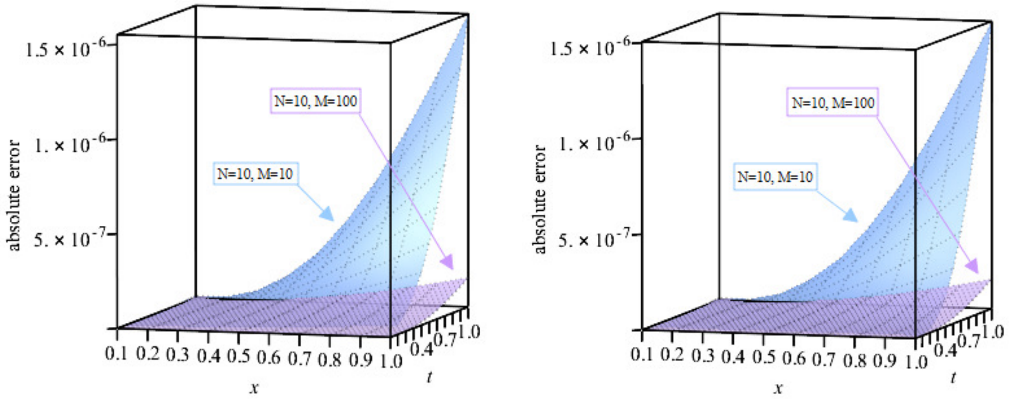

Figure 7: Comparison of absolute error , between $N = 10, M = 10$ and $N = 10, M = 100$ by Lobatto IIIC method with $m = 2$ (left) and $m = 3$ (right) for Example 2.

Table 11: Maximum value of errors for Example 2 by Radau IIA method with m =2, 3 and Lobatto IIIC method with m =2, 3, 4.

	Radau IIA method		Lobatto IIIC method		
(N,M)	$m = 2$	$m = 3$	$m = 2$	$m = 3$	$m = 4$
$(10,10)$	$0.1e-5$	$0.1e-5$	$0.1e-4$	$0.1e-5$	$0.1e-5$
$(10,100)$	$0.1e-6$	$0.1e-6$	$0.3e-6$	$0.1e-6$	$0.1e-6$

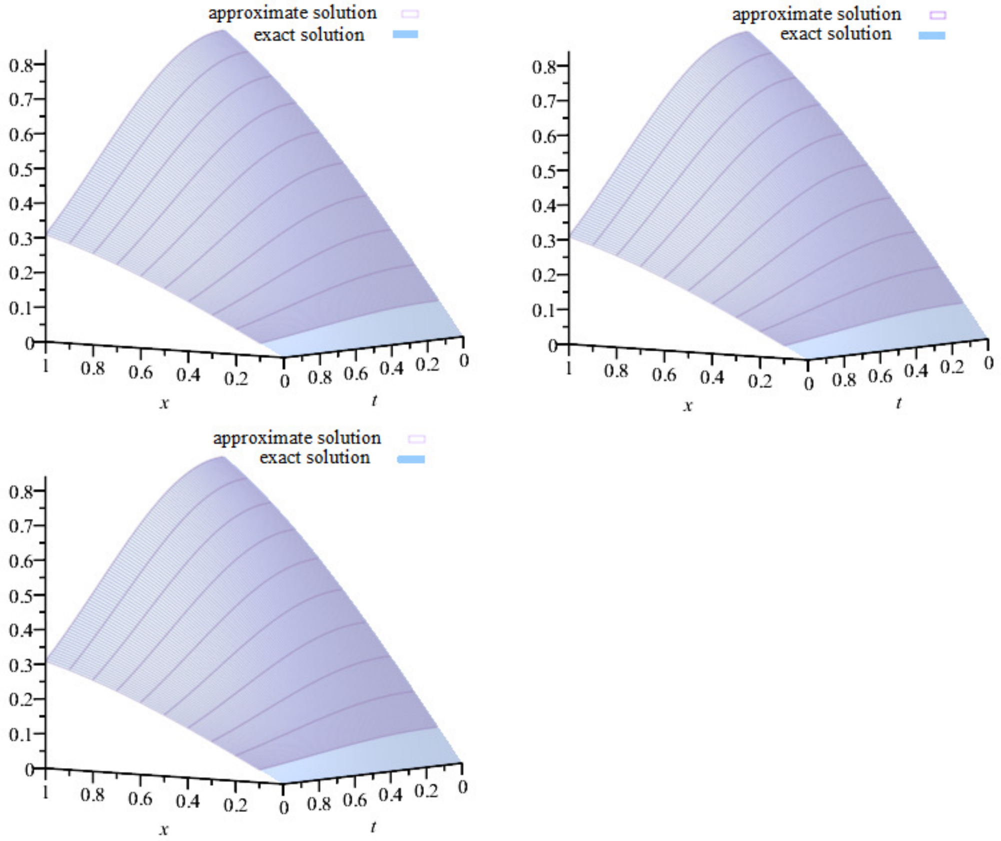

Figure 8: Comparison of exact and approximate solutions by the semi-analytical method with $N = 10$ (row:1, column:1), Lobatto IIIC method with $m = 4, N = 10, M = 100$ (row:1, column:2) and Radau IIA method with $m = 3, N = 10, M = 100$ (row:2, column:1) for Example 2.

Acknowledgments

The authors are very grateful to the anonymous referees and the editor for carefully reading the paper and for their constructive comments and suggestions that improved the early version of this paper.

References

[1] W.A. Adkins, M.G. Davidson, *Ordinary Differential Equations*, Springer Science Business Media New York, 2012.

[2] S. Abbas, M. Benchohra, G.M. N'Guerekata, *Advanced fractional differential and integral equations*, Nova Science Publishers, Incorporated, 2014.

[3] R.P. Agarwal, B. de Andrade, G, Siracusa, *On fractional integro-differential equations with state-dependent delay*, Comput. Math. Appl. **62** (2011) 1143–1149.

[4] K. Burrage, *Stability and Efficiency Properties of Implicit Runge-Kutta Methods*, Ph.D. Thesis, University of Auckland, 1978.

[5] J.C. Butcher, *On the implementation of implicit Runge-Kutta methods*, BIT Numer. Math. **16** (1976) 237–240.

[6] K. Diethelm, *The Analysis of Fractional Differential Equations: An Application-Oriented Exposition Using Differential Operators of Caputo Type*, Springer Science Business Media, London, New York, 2010.

[7] R. Gorenflo, F. Mainardi, *Fractals and Fractional Calculus in Continuum Mechanics*, Springer Vienna, Vienna, 1997.

[8] A.R. Hadhoud, A.A. Rageh, P. Agarwal, *Numerical method for solving twodimensional of the space and spacetime fractional coupled reactiondiffusion equations*, Math. Methods Appl. Sci., 2022, doi: https://doi.org/10.1002/mma.8891.

[9] G. Hall, J.M. Watt , *Modern Numerical Methods for Ordinary Differential Equations*, Clarendon Press, Oxford, 1976.

[10] A.A. Kilbas, H.M. Srivastava, J.J. Trujillo, *Theory and Applications of Fractional Differential Equations*, Elsevier, Neterlands, 2006.

[11] A.A. Khajehnasiri, R. Ezzati, *Boubaker polynomials and their applications for solving fractional two-dimensional nonlinear partial integro-differential Volterra integral equations*, Comp. Appl. Math. **41** (2022) 82.

[12] D.D. Kushal, J. Singh, *Numerical solution of two-dimensional fractional-order reaction advection sub-diffusion equation with finite-difference Fibonacci collocation method*, Math. Comput. Simul. **181** (2021) 38–50.

[13] A.C.J. Luo, *Memorized Discrete Systems and Time-Delay*, Springer International Publishing, 2017.

[14] F. Mirzaee, S. Alipour, N. Samadyar, *A numerical approach for solving weakly singular partial integrodifferential equations via twodimensionalorthonormal Bernstein polynomials with the convergence analysis*, Numer. Methods Partial Differ. Equ.**35** (2019) 615–637.

[15] M. Mojahedfar, Abolfazl Tari, S. Shahmorad, *A computational method for time fractional partial integro-differential equations*, J. Appl. Anal. **26** (2020) 315–323.

[16] M. Mojahedfar, A. Tari, *A numerical method for partial fractional Fredholm integro-differential equations*, Comput. Methods Deiffer. Equ. **7** (2019) 152–162.

[17] A. Prothero, A. Robinson, *On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations*, Math. Comput.**28** (1974) 145–162.

[18] N. Samadyar, F. Mirzaee, *Numerical scheme for solving singular fractional partial integrodifferential equation via orthonormal Bernoulli polynomials*, Int. J. Numer. Model.: Electron. Netw. Devices Fields, **32**, (2019) 1–18.

[19] S. Saha Ray, S. Behera, *Comparison of Two Wavelet Methods for Accurate Solution of Two-Dimensional Nonlinear Volterra Integral Equation*, Iran. J. Sci. Technol. Trans. A: Sci. **45** (2021) 2091–2108.

[20] L. Saeedi, A. Tari, E. Babolian, *A Study on functional fractional integro-differential equations of Hammerstein type*, Comput. Methods Differ. Equ. **8** (2020) 173–193.

[21] W.E. Schiesser, *The Numerical Method of Lines*, Academic Press, New York, 1991.

[22] H.M. Srivastava, R.K. Parmar, P. Chopra, *A class of extended fractional derivative operators and associated generating relations involving hypergeometric functions*, Axioms **1** (2012) 238–258.

[23] V.E. Tarasov, *Fractional integro-differential equations for electromagnetic waves in dielectric media*, Theor. Math. Phys. **158** (2009) 355–359.

[24] Z. Zlatev, *Modified diagonally implicit RungeKutta methods*, SIAM J. Sci. Statist. Comput. **2** (1981) 321–334.