Original Research

A Class of Block Multi-derivative Numerical Techniques for Singular Advection Equations

M. O. Ogunniran*

Department of Mathematical Sciences, Osun State University Osogbo, Nigeria

Abstract

The integration of some differential equations is hard to acquire because of the presence of singular point(s) in these equations. These equations are best solved by some unique technique. Multi-derivative techniques have a long history of powerful integration of such equations yet till date, a couple of class of this technique has been explored for integrating partial differential equations. This work centers around the development, analysis and implementation of a class of multi-derivative technique on partial differential equations. The approaches were effectively analyzed and were turned out to be consistent, stable and convergent. Numerical outcomes got likewise demonstrated the approximation quality of the technique over existing techniques in literature.

Keywords: Advection equations, Singular point, Multi-derivative, Conservation law, off-step point

1. Introduction

Considering an equation which is known as the linear advection equation for a quantity $u(x, t)$:

$$\frac{\partial u}{\partial t} + v(x, t)\frac{\partial u}{\partial x} = 0 \tag{1}$$

where $v(x, t)$ is known, This type of equation in (1) is classified as first order hyperbolic partial differential equation. Equation (1) is termed as a conservation law since it expresses conservation of mass, energy or momentum under the conditions for which it is derived, i.e. the assumptions on which the equation is based.

In physical phenomenon, $v(x, t)$ is linear or flow velocity. Although equation (1) is possibly the simplest partial differential equation, this simplicity is deceptive in the sense that it can be very difficult to integrate numerically due to the presence of singularity, a distinctive feature of first order hyperbolic partial differential equations.

Equation (1) will have to be supplemented with an initial condition:

$$u(x, 0) = u_0(x) \tag{2}$$

and if necessary boundary condition(s):

$$\begin{aligned} u(a, t) &= u_1(t), \\ u(b, t) &= u_2(t). \quad x \in [a, b] \end{aligned} \tag{3}$$

One of the most popular techniques applied to the solution of Advection equation is finite difference method [1,2]. In recent times, the finite element and the finite volume methods were

*Corresponding Author Tel. No: +2347039104606
Email address:* muideen.ogunniran@uniosun.edu.ng (M. O. Ogunniran)

also introduced. Many special methods have also been developed to deal with Singularly Perturbed Problems. Many Authors [3,4,5,6] are devoted to Singularly Perturbed Problems of ordinary differential equation and the authors discussed the situation and width of boundary layer(s) and give some effective numerical algorithms. In the work of Xianyl [7], a linear hybrid variable method was developed for the solution of Advection equations and an efficient numerical treatment of non-linearities in the advection-diffusion-reaction equations was presented by Utku et al. [8]. An extensive work was done by Gaetamo and Silvia [9] on the fractional diffusion-advection equations for fluids and plasmas. Adrian [10] presented a generalization of prefactored compact schemes for the solution of advection equations.

2. Method and Analysis

The methods were derived by assuming a continuous approximant for $u_n(x)$ of a multi-step multi-derivative method of the form:

$$u_n(x) = \alpha_n(x)y_n + \sum_{i=1}^{l} h^i(\beta_{i,0}(x)f_n^{(i-1)} + \beta_{i,v}(x)f_{n+v}^{(i-1)})$$
$$+ \sum_{j=1}^{k} \beta_{j,k}(x)f_{n+j}^{(i-1)}) \approx y(x) \tag{4}$$

for solving IVP: $y' = f(x, y)$, $y(0) = y_0$, where $f(x, y)$ is continuous and j^{th} differentiable, y_n is an approximation to $y(x_n)$, $x_n = nh$, $h > 0$ and $f_m^{(j)} = f^{(j)}(x_m, y_m)$ such that:

$$f^{(0)}(x, y) = f(x, y),$$
$$f^{(j)}(x, y) = \frac{\partial f^{(j-1)}(x, y)}{\partial x} + f(x, y)\frac{\partial f^{(j-1)}(x, y)}{\partial y},$$

where k is the step number, l is derivative order, $v \in (0, k)$ is an off-step point and j is the derivative order of $f(x, y)$. However, the continuous coefficients $\beta_{i,j}(x)$, $i = 0, v, k$, $j = 1(1)k$ were determined.
To this end, approximation of the exact solution $y(x)$ was sought by evaluating the function:

$$u(x) = \sum_{j=0}^{r+ls-1} a_j x^j \tag{5}$$

where a_j, $(j = 0, 1, ..., r+ls-1)$ are coefficients determined, x^j are the basis functions of degree $r + ls - 1$, l, r and s being the derivative order, interpolating and collocations points respectively.
While ensuring that the function (4) corresponds with the analytical solution at the end point x_n, the following conditions were imposed on $u(x)$ and its derivatives $u^{(k)}(x)$ to get the coefficients of the desired methods:

$$\left.\begin{aligned}u(x_{n+j}) &= y_{n+j}, \quad j = 0 \\ u'(x_{n+j}) &= f_{n+j}, \quad j \in [0, \cdots, k] \\ u''(x_{n+j}) &= f'_{n+j} = g_{n+j}, \quad j \in [0, \cdots, k] \\ u'''(x_{n+j}) &= f''_{n+j} = h_{n+j}, \quad j \in [0, \cdots, k] \\ &\vdots \\ u^{(k)}(x_{n+j}) &= f_{n+j}^{(k-1)}, \quad j \in [0, \cdots, k]\end{aligned}\right\} \tag{6}$$

The coefficients obtained were substituted into (5) to obtain the continuous coefficients in (4) which were then evaluated at x_{n+k} to obtain the desired methods.

2.1. One-step, First Derivative and Two off-step Method

Here, the first derivative evaluation was considered taking into consideration, one step and two off-step points. The procedure is as follows:
For $k = 1$, $l = 1$ and $v = \frac{1}{7}, \frac{2}{7}$ (4) becomes:

$$u(x) = \alpha_n(x)y_n + h(\beta_{1,0}(x)f_n + \beta_{1,\frac{1}{7}}(x)f_{n+\frac{1}{7}}$$
$$+ \beta_{1,\frac{2}{7}}(x)f_{n+\frac{2}{7}} + \beta_{1,1}(x)f_{n+1}) \tag{7}$$

while taking $r = 1$ and $s = 4$, (5) and its derivative become:

$$u(x) = \sum_{j=0}^{4} a_j x^j \tag{8}$$

$$u'(x) = \sum_{j=1}^{4} j a_j x^{j-1} \tag{9}$$

Imposing the following conditions obtained from (6):

$$u(x_{n+j}) = y_{n+j}, \quad j = 0. \tag{10}$$
$$u'(x_{n+j}) = f_{n+j}, \quad j = 0, \frac{1}{7}, \frac{2}{7}, 1 \tag{11}$$

a system of five algebraic equations with five unknowns were obtained. The 5×5 system of equations obtained are given below:

$$a_0 = y_n \tag{12}$$
$$a_1 = f_n \tag{13}$$
$$a_1 + \frac{2}{7}a_2 + \frac{3}{49}a_3 + \frac{4}{343}a_4 = f_{n+\frac{1}{7}} \tag{14}$$
$$a_1 + \frac{4}{7}a_2 + \frac{12}{49}a_3 + \frac{32}{343}a_4 = f_{n+\frac{2}{7}} \tag{15}$$
$$a_1 + 2a_2 + 3a_3 + 4a_4 = f_{n+1}. \tag{16}$$

Solving the resulting equations gives:

$$a_0 = y_n \tag{17}$$
$$a_1 = f_{n+1} \tag{18}$$
$$a_2 = -\frac{23}{4}f_n + \frac{49}{6}f_{n+\frac{1}{7}} - \frac{49}{20}f_{n+\frac{2}{7}} + \frac{1}{30}f_{n+1} \tag{19}$$
$$a_3 = \frac{35}{3}f_n - \frac{49}{2}f_{n+\frac{1}{7}} + \frac{196}{15}f_{n+\frac{2}{7}} - \frac{7}{30}f_{n+1} \tag{20}$$
$$a_4 = -\frac{343}{40}f_{n+\frac{2}{7}} + \frac{49}{120}f_{n+1} - \frac{49}{8}f_n + \frac{343}{24}f_{n+\frac{1}{7}} \tag{21}$$
$$a_4 = -\frac{49}{8}f_n + \frac{343}{24}f_{n+\frac{1}{7}} - \frac{343}{40}f_{n+\frac{2}{7}} + \frac{49}{120}f_{n+1} \tag{22}$$

Substituting these values in (7) and collecting like terms in y_n, f_n, $f_{n+\frac{1}{7}}$, $f_{n+\frac{2}{7}}$, f_{n+1} yields the continuous method:

$$y(x) = \alpha_n(x)y_n + h(\beta_{0,1}(x)f_n + \beta_{1,\frac{1}{7}}(x)f_{n+\frac{1}{7}}$$
$$+ \beta_{1,\frac{2}{7}}(x)f_{n+\frac{2}{7}} + \beta_{1,1}(x)f_{n+1}) \tag{23}$$

where:

$$\alpha_n(x) = 1$$
$$\beta_{1,0}(x) = -\frac{49(2x-1)^4}{8} + \frac{35(2x-1)^3}{3} - \frac{23(2x-1)^2}{4} + 2x - 1$$
$$\beta_{1,\frac{1}{7}}(x) = \frac{343(2x-1)^4}{24} - \frac{49(2x-1)^3}{2} + \frac{49(2x-1)^2}{6}$$
$$\beta_{1,\frac{2}{7}}(x) = -\frac{343(2x-1)^4}{40} + \frac{196(2x-1)^3}{15} - \frac{49(2x-1)^2}{20}$$
$$\beta_{1,1}(x) = \frac{49(2x-1)^4}{120} - \frac{7(2x-1)^3}{30} + 1/30\,(2x-1)^2 \quad (24)$$

Evaluating (23) at x_{n+1} yields the one-step, first derivative main method possessing two off-grid points:

$$y_{n+1} = y_n + \frac{h}{24}(19f_n - 49f_{n+\frac{1}{7}} + 49f_{n+\frac{2}{7}} + 5f_{n+1}) \quad (25)$$

Also, evaluating (23) at $x_{n+\frac{1}{7}}$ and $x_{n+\frac{2}{7}}$ yields the additional method as:

$$y_{n+\frac{1}{7}} = y_n + \frac{h}{5880}(335f_n + 595f_{n+\frac{1}{7}} - 91f_{n+\frac{2}{7}} + f_{n+1})$$
$$y_{n+\frac{2}{7}} = y_n + \frac{h}{21}(f_n + 4f_{n+\frac{1}{7}} + f_{n+\frac{2}{7}}) \quad (26)$$

The one-step, first derivative and two off-grid points method is presented in block form as follows:

$$y_{n+1} = y_n + \frac{h}{24}(19f_n - 49f_{n+\frac{1}{7}} + 49f_{n+\frac{2}{7}} + 5f_{n+1})$$
$$y_{n+\frac{1}{7}} = y_n + \frac{h}{5880}(335f_n + 595f_{n+\frac{1}{7}} - 91f_{n+\frac{2}{7}} + f_{n+1})$$
$$y_{n+\frac{2}{7}} = y_n + \frac{h}{21}(f_n + 4f_{n+\frac{1}{7}} + f_{n+\frac{2}{7}}) \quad (27)$$

2.2. Two-step, First Derivative and Two off-step Method

Here, a new member of the methods is obtained. The procedure is as follows:
For $k = 2$, $l = 1$, and $v = \frac{1}{7}$ and $\frac{9}{7}$, (4) becomes:

$$u(x) = \alpha_n(x)y_n + h(\beta_{1,0}(x)f_n + \beta_{1,\frac{1}{7}}(x)f_{n+\frac{1}{7}}$$
$$+ \beta_{1,\frac{9}{7}}(x)f_{n+\frac{9}{7}} + \beta_{1,1}(x)f_{n+1} + \beta_{1,2}(x)f_{n+2}) \quad (28)$$

taking $r = 1$ and $s = 5$, (5) and its derivative become:

$$u(x) = \sum_{j=0}^{5} a_j x^j \quad (29)$$

$$u'(x) = \sum_{j=1}^{5} j a_j x^{j-1} \quad (30)$$

Imposing the following conditions obtained from (6):

$$u(x_{n+j}) = y_{n+j}, \quad j = 0. \quad (31)$$

$$u'(x_{n+j}) = f_{n+j}, \quad j = 0, \frac{1}{7}, 1, \frac{9}{7}, 2. \quad (32)$$

The two-step, first derivative and two off-grid point method is presented in block form as follows:

$$y_{n+2} = y_n + h(-\frac{53}{135}f_n + \frac{2401}{2340}f_{n+\frac{1}{7}} + \frac{11}{4}f_{n+1}$$
$$+ \frac{2401}{2700}f_{n+\frac{9}{7}} + \frac{227}{975}f_{n+2})$$
$$y_{n+\frac{1}{7}} = y_n + h(\frac{3391}{52920}f_n + \frac{10579}{131040}f_{n+\frac{1}{7}}$$
$$- \frac{41}{8820}f_{n+1} + \frac{439}{151200}f_{n+\frac{9}{7}} - \frac{79}{382200}f_{n+2})$$
$$y_{n+1} = y_n + h(-\frac{257}{1080}f_n + \frac{14749}{18720}f_{n+\frac{1}{7}} + \frac{127}{180}f_{n+1}$$
$$- \frac{5831}{21600}f_{n+\frac{9}{7}} + \frac{113}{7800}f_{n+2})$$
$$y_{n+\frac{9}{7}} = y_n + h(-\frac{3141}{13720}f_n + \frac{11259}{14560}f_{n+\frac{1}{7}}$$
$$+ \frac{5913}{6860}f_{n+1} - \frac{747}{5600}f_{n+\frac{9}{7}} + \frac{11421}{891800}f_{n+2}) \quad (33)$$

2.3. Two-step, Second Derivative and One off-step Method

Here $k = 2$, $l = 2$, and $v = \frac{1}{7}$ is the chosen off-step point. The new approximation now becomes:

$$u(x) = \alpha_n(x)y_n + h(\beta_{1,0}(x)f_n + \beta_{1,\frac{1}{7}}(x)f_{n+\frac{1}{7}}$$
$$+ \beta_{1,1}(x)f_{n+1} + \beta_{1,2}(x)f_{n+2})$$
$$+ h^2(\beta_{2,0}(x)f_n + \beta_{2,\frac{1}{7}}(x)g_{n+\frac{1}{7}}$$
$$+ \beta_{2,1}(x)g_{n+1} + \beta_{2,2}(x)g_{n+2}) \quad (34)$$

where $g(x_{n+j}) = f'(x_{n+j}) = u''(x_{n+j}), \quad j = 0, \cdots, k$.
Setting $r = 1$ and $s = 4$, the interpolating function now becomes:

$$u(x) = \sum_{j=0}^{8} a_j x^j. \quad (35)$$

Obtaining the first and second derivatives, we have:

$$u'(x) = \sum_{j=1}^{8} j a_j x^{j-1}$$
$$u''(x) = \sum_{j=2}^{8} j(j-1)a_j x^{j-2}. \quad (36)$$

Imposing the following conditions as obtained from (6):

$$u(x_{n+j}) = y_{n+j}, \quad j = 0 \quad (37)$$

$$u'(x_{n+j}) = f_{n+j}, \quad j = 0, \frac{1}{7}, 1, 2. \quad (38)$$

$$u''(x_{n+j}) = g_{n+j}, \quad j = 0, \frac{1}{7}, 1, 2. \quad (39)$$

Thus, the required method is given in block method as:

$$y_{n+2} = -\frac{31h^2 g_{n+2}}{845} + \frac{16h^2 g_{n+1}}{45} + \frac{19208h^2 g_{n+\frac{1}{7}}}{7605}$$
$$+ \frac{7}{5}h^2 g_n + \frac{781h f_{n+2}}{2197} + \frac{4}{9}h f_{n+1}$$
$$- \frac{470596h f_{n+\frac{1}{7}}}{19773} + 25h f_n + y_n$$

$$y_{n+\frac{1}{7}} = -\frac{15997h^2 g_{n+2}}{66805808160} - \frac{33203h^2 g_{n+1}}{3557705760}$$
$$- \frac{516419h^2 g_{n+\frac{1}{7}}}{250417440} + \frac{565619h^2 g_n}{395300640}$$
$$+ \frac{191117h f_{n+2}}{173695101216} + \frac{9703h f_{n+1}}{304946208}$$
$$+ \frac{3016667h f_{n+\frac{1}{7}}}{39862368} + \frac{5308663h f_n}{79060128} + y_n$$

$$y_{n+1} = -\frac{43h^2 g_{n+2}}{81120} - \frac{197h^2 g_{n+1}}{4320}$$
$$+ \frac{333739h^2 g_{n+\frac{1}{7}}}{730080} + \frac{101h^2 g_n}{480}$$
$$+ \frac{539h f_{n+2}}{210912} + \frac{919h f_{n+1}}{2592}$$
$$- \frac{19176787h f_{n+\frac{1}{7}}}{5694624}$$
$$+ \frac{385h f_n}{96} + y_n \quad (40)$$

3. Analysis of Methods

3.1. Order and Error Constant

In this section, analysis of the methods using Taylor's series method for the derivation of order and error constant of the methods were carried out. Following Lambert (1991) and Fatunla (1991), the linear difference operator $\mathcal{L}$ associated with the method is defined by:

$$\mathcal{L}[y(x); h] = \sum_{i=0}^{k} \alpha_i(x) y(x + ih)$$
$$- \sum_{i=1}^{l} h^i (\beta_{i,0}(x) y^{(i)}(x + ih)$$
$$+ \beta_{i,v}(x) y^{(i)}(x + vh)$$
$$+ \sum_{j=1}^{k} \beta_{j,k}(x) y^{(j)}(x + jh)) \qquad (41)$$

where $y(x)$ is an arbitrary function, continuously differentiable on $[a, b]$.

Recall that; while expanding $y(x + ih)$ and its derivatives $y^{(j)}(x + ih)$ as Taylor's series about x, and collecting terms gives:

$$\mathcal{L}[y(x); h] = c_0 y(x) + c_1 h y^{(1)}(x) + \cdots + c_q h^q y^{(q)}(x) + \cdots (42)$$

where $c_q, \ q = 0, 1, 2, \cdots$ are the constant coefficients are given as follows:

$$\left.\begin{array}{c} c_0 = 1 - \alpha_n \\ c_1 = k - \beta_{i,0} - \beta_{i,v} - \sum_{j=1}^{k} \beta_{j,k} \\ \vdots \\ c_q = \dfrac{k^q}{q!} - \dfrac{v^{q-1}}{(q-1)!} \beta_{i,v} - \dfrac{k^{q-1} + 1}{(q-1)!} \sum_{j=1}^{k} \beta_{j,k} \end{array}\right\} \qquad (43)$$

According to Henrici (1962), the main method (4) has order p if:

$$\mathcal{L}[y(x); h] = O(h^{p+1}), \qquad (44)$$

where:

$$c_0 = c_1 = \cdots = c_p = 0, \quad c_{p+1} \neq 0.$$

Therefore, c_{p+1} is the error constant and $c_{p+1}h^{p+1}(x_n)$ is the principal local truncation error at point x_n.

3.2. Consistency and Zero-stability

3.2.1. Consistency

According to Lambert (1991), a numerical scheme is said to be consistent if the order p of the method is such that $p > 1$.

3.2.2. Zero Stability

A numerical method is said to be zero-stable if no root of the first characteristic polynomial has modulus greater than one that is if every root with modulus one is simple (Lambert, 1991).
A Linear Multi-step Method is said to be zero-stable if no root of the first characteristic polynomial defined by:

$$\rho(r) = det[rA^{(0)} - A^{(1)}] \qquad (45)$$

where $A^{(0)}$ is an identity matrix of dimension k, $A^{(1)}$ is a matrix of dimension k and satisfy $|r_s| \leq 1$ and every root of $|r_s| = 1$, $s = 1(1)k$, has multiplicity not exceeding as $h \to 0$.

In what follows, the k-step, m-derivative block method is written as a matrix finite difference equation of the form:

$$\begin{aligned} A^{(0)}Y_{\mu+1} =\ & A^{(1)}Y_{\mu} + h[B^{(0)}F_{\mu+1} + B^{(1)}F_{\mu}] \\ & + h^2[C^{(0)}G_{\mu+1} + C^{(1)}G_{\mu}] \\ & + \cdots + h^m[Z^{(0)}Z_{\mu+1} + Z^{(1)}Z_{\mu}] \end{aligned} \qquad (46)$$

where m is the derivative order. $Y_{\mu+1}$, Y_{μ}, $F_{\mu+1}$, F_{μ}, $G_{\mu+1}$, G_{μ}, $\cdots$, $Z_{\mu+1}$, Z_{μ} are column matrix determined from the derived methods. $A^{(0)}$, $A^{(1)}$, $B^{(0)}$, $B^{(1)}$, $C^{(0)}$, $C^{(1)}$, $\cdots$, $Z^{(0)}$, $Z^{(1)}$ are coefficients matrices of the derived methods. To this end, the zero stability of the derived methods are discussed below. For method (27), the matrix difference equation is:

$$A^{(0)}Y_{\mu+1} = A^{(1)}Y_{\mu} + h[B^{(0)}F_{\mu+1} + B^{(1)}F_{\mu}] \qquad (47)$$

where:

$$\left.\begin{array}{c} Y_{\mu+1} = (y_{n+\frac{1}{7}}, y_{n+\frac{2}{7}}, y_{n+1})^T \\ Y_{\mu} = (y_{n-\frac{1}{7}}, y_{n-\frac{2}{7}}, y_n)^T \\ F_{\mu+1} = (f_{n+\frac{1}{7}}, f_{n+\frac{2}{7}}, f_{n+1})^T \\ F_{\mu} = (f_{n-\frac{1}{7}}, f_{n-\frac{2}{7}}, f_n)^T \\ \\ A^{(0)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \ A^{(1)} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \\ \\ B^{(0)} = \begin{pmatrix} \frac{595}{5880} & -\frac{91}{5880} & \frac{1}{5880} \\ \frac{4}{21} & \frac{1}{21} & 0 \\ -\frac{49}{24} & \frac{49}{24} & \frac{5}{24} \end{pmatrix}, \ B^{(1)} = \begin{pmatrix} 0 & 0 & \frac{335}{5880} \\ 0 & 0 & \frac{1}{21} \\ 0 & 0 & \frac{19}{24} \end{pmatrix} \end{array}\right\} \qquad (48)$$

Zero-stability is a concept concerned with the limiting factor of the difference equation as $h \to 0$. Therefore, the difference equation is reduced to:

$$A^{(0)}Y_{\mu+1} = A^{(1)}Y_{\mu} \qquad (49)$$

in which the first characteristic polynomial:

$$\rho(r) = 0. \qquad (50)$$

The solution of equation (50) corresponds to equation (45), thus solving (50) with parameters obtained in (48) gives the roots of r.

$$\rho(r) = det\left[r \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \right] = 0 \qquad (51)$$

$$\rho(r) = det\left(\begin{pmatrix} r & 0 & -1 \\ 0 & r & -1 \\ 0 & 0 & r-1 \end{pmatrix} \right) = 0 \qquad (52)$$

from (52) we have:

$$\begin{aligned} r^2(r - 1) &= 0 \\ r &= 0, 0, 1. \end{aligned} \qquad (53)$$

Table 1: Table of Orders and Error Constants of the Derived Multi-Derivative Methods

Method	Evaluating points	Order	Error Constant
27	x_{n+1}	4	$\frac{-31}{35280} = -8.79E - 04$
	$x_{n+\frac{1}{7}}$	4	$\frac{-1}{246960} = -4.05E - 06$
	$x_{n+\frac{2}{7}}$	4	$\frac{-1}{1512630} = -6.61E - 07$
33	x_{n+2}	5	$-\frac{2}{1575} = 1.27E - 03$
	$x_{n+\frac{1}{7}}$	5	$\frac{2491}{282357600} = 8.82E - 06$
	x_{n+1}	5	$-\frac{53}{117600} = 4.51E - 04$
	$x_{n+\frac{9}{7}}$	5	$-\frac{39447}{94119200} = 4.09E - 04$
40	x_{n+2}	8	$\frac{157}{38896200} = 4.04E - 06$
	$x_{n+\frac{1}{7}}$	8	$\frac{162133}{1025046183571200} = 1.58E - 10$
	x_{n+1}	8	$\frac{307}{1244678400} = 2.47E - 07$

Since no root of the characteristic equation violates the condition: $|r_s| \leq 1$, $s = 1(1)m$ and m is the highest power of the characteristics equation $\rho(r) = 0$. It is concluded that method (27) is zero-stable.

Table 2: Table of roots of the characteristic equations for the Derived Multi-Derivative Methods

Method	Absolute value of Roots, $|r_s|$
27	0,0,1.
33	0,0,0,1.
40	0,0,1.

3.3. Convergence

Stating without proof and following the fundamental theorem of Dahliquist as reported by Henrici (1962) for a linear multi-step method.

3.3.1. Theorem 1

The necessary and sufficient conditions for a numerical method to be convergent are that it is consistent and zero-stable. Here, it is sufficient to say that the derived methods are convergent since they are all consistent and zero-stable.

3.4. Linear Stability

The linear stability properties of the derived methods are determined by expressing them in a form applicable to the test problem:

$$y' = \lambda y, \quad \text{for which } y'_n = \lambda y_n, \quad y''_n = \lambda^2$$
$$y_n, \cdots, y_n^{(n)} = \lambda^n y_n., \quad \lambda < 0 \tag{54}$$

to yield:

$$Y_{\mu+1} = M(z)Y_\mu, \quad z = h\lambda \tag{55}$$

where the amplification matrix $M(z)$ is given by:

$$M(z) = (A^{(0)} - zB^{(0)} - z^2C^{(0)} - \cdots - z^lZ^{(0)})^{-1}(A^{(1)}$$
$$+ zB^{(1)} + z^2C^{(1)} + \cdots + z^lZ^{(1)}). \tag{56}$$

where $z = h\lambda$. The matrix $M(z)$ has eigenvalues $\xi_1, \xi_2, \cdots, \xi_m = 0, 0, \cdots, \xi_m$, where the dominant eigenvalue ξ_m is the stability

function $R(z)$ which is a rational function with real coefficients, m is the order of $R(z)$. The stability function and plot for each of the derived methods are as given below:
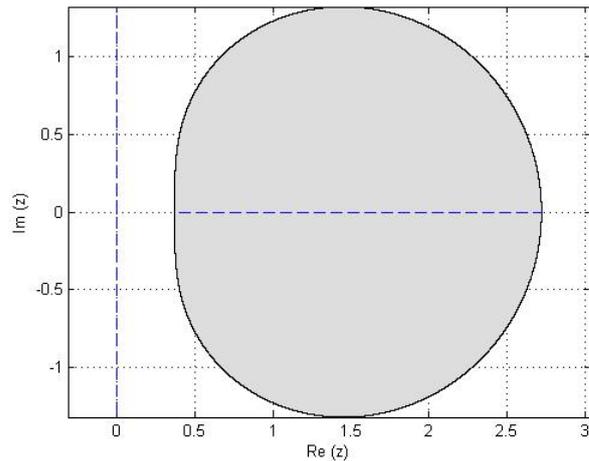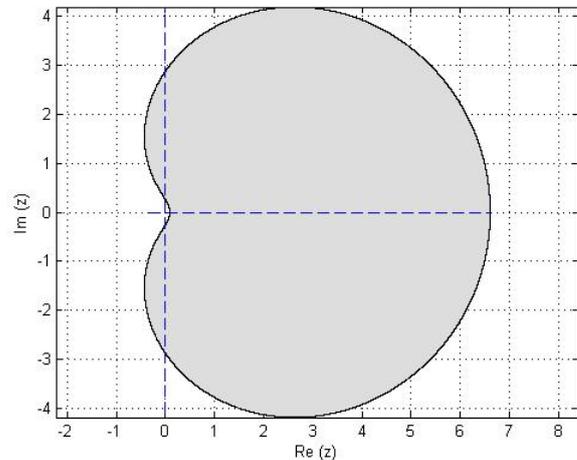

Figure 1: Stability Region of Method (27)

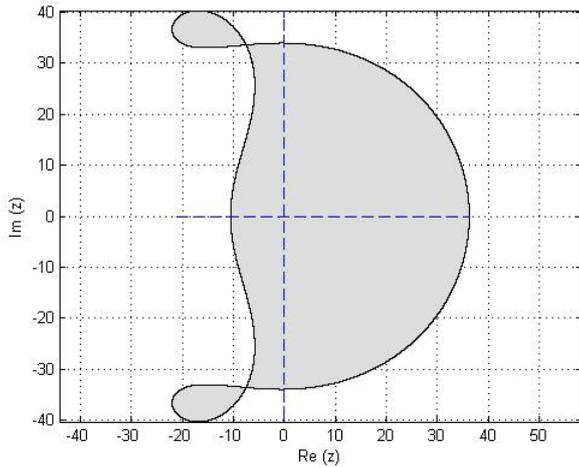Figure 2: Stability Region of Method (33)

Figure 3: Stability Region of Method (40)

4. Numerical Examples and Results

5. Computational Details

In the implementation of the derived methods, a system of nonlinear equations must be solved in order to obtain the desired approximation. To solve these nonlinear systems, a Newton-Krylov solver, nsoli.m or a modified Newton solver, nsold was used and compared numerical results with standard and existing methods in literature. It is important to point that the numerical methods were programmed via MATLAB 9.2 version on a personal computer with the following specifications:

- System name- HP Pavilion x360 Convertible

- Processor- Intel(R) Core(TM) i3-7100U CPU @ 2.40GHz

- Installed memory (RAM)- 8.00GB

- System Type- 64-bits Operating System, x64-based processor

- Operating system- Windows 10

Moreso, computational experiments were done with software optimization and only the points of emphases were shown.

5.0.1. Example 1

Advection equation where $v(x, t)$ is constant:

$$\frac{\partial u}{\partial t} = -v\frac{\partial u}{\partial x}$$
$$u(x, 0) = \exp(-\frac{x^2}{L^2})$$
Exact Solution: $u(x, t) = \exp(-\frac{(x-vt)^2}{L^2})$
$$v = 0.5, L = 10, x \in [0, 1], t \in [0, 1] \qquad (57)$$

is considered.

5.0.2. Example 2

An advection equation where $v(x, t) = \dfrac{1 + x^2}{1 + 2xt + 2x^2 + x^4}$

$$\frac{\partial u}{\partial t} = -v(x, t)\frac{\partial u}{\partial x}$$
$$u(x, 0) = \exp(-\frac{x}{L^2})$$
Exact Solution: $u(x, t) = \exp(-\frac{1}{L^2}(x - \frac{t}{1+x^2}))$
$$L = 10, x \in [0, 1], t \in [0, 1] \qquad (58)$$

is also considered.

5.0.3. Example 3

The inviscid Burgers equation where $v(x, t) = u(x, t)$

$$\frac{\partial u}{\partial t} = -u(x, t)\frac{\partial u}{\partial x}$$
$$u(x, 0) = x$$
Exact Solution: $u(x, t) = \dfrac{x}{t + 1}$
$$x \in [0, 1], t \in [0, 1]. \qquad (59)$$

5.0.4. Example 4

A nonlinear advection equation with perturbation term:

$$\left.\begin{array}{r}\epsilon\dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} = f(x, t), \quad (x, t) \in D = [0, 1], \times[0, 1]\\ u(0, t) = \epsilon t, \quad u(x, 0) = x,\end{array}\right\}(60)$$

where $f(x, t) = \epsilon^2 + \epsilon t + x$, $u(x, t) = \epsilon t + x$ is the true solution, $\epsilon = 10^{-4}$ is also considered.

6. Tables of Results and Comparison

In this section, the standards methods of Upwind, Lax method, the method of Wang *et al.* [11] and the Method were employed successfully for solving the advection equations. The numerical results of the proposed method show some superiority over other methods in-terms of accuracy and reliability. Moreover, the Method is also effective for solving the equation with singular terms and other nonlinear singular perturbation problems. Wang *et al.* [11] used a perturbation method and reproducing kernel method in solving a class of singularly perturbed partial differential equation. From Table 3 above, it could be observed that the Method shows high computational strength as it compares favourably well with some known standard methods reported. [1]

From the table above, it could also be observed that the Method shows high computational strength as it compares favourably well with some known standard methods reported. [2]

The superiority of the proposed methods could be established in the table above. From the table above, it could also be

----

[1](x, t) is the coordinate of x and t, $u(x, t)$ is the exact solution, UM is the Upwind method, LM is the Lax method, Method represents the solution using the Method and Relative error=$| u(x, t) -$ Method $|/u(x, t)$. Percentage error = Relative error × 100%. CPU time is the method's computational time.

[2](x, t) is the coordinate of x and t, $u(x, t)$ is the exact solution, UM is the Upwind method, LM is the Lax method, Method represents the solution using the Method and Relative error=$| u(x, t) -$ Method $|/u(x, t)$. Percentage error = Relative error × 100%. CPU time is the method's computational time.

Table 3: Numerical Comparison for Example 1 for $v(x,t) = 0.5$, $\Delta x = 0.1$, $\Delta t = 0.05$, $L = 10$

(x,t)	$u(x,t)$	UM	LM	Method (27)	Relative error	Percentage error (%)
(0.1000,0.1000)	0.9999750003	0.9999453153	0.9998539643	0.9999937500	1.87502 E -05	0.0019
(0.2000,0.2000)	0.9999000050	0.9998334259	0.9996180921	0.9999000050	0.00000	0.00
(0.3000,0.3000)	0.9997750253	0.9996699630	0.9993277560	0.9996000800	1.74985 E -04	0.017
(0.4000,0.4000)	0.9996000800	0.9994562666	0.9989757869	0.9991004049	4.99875 E -04	0.050
(0.5000,0.5000)	0.9993751953	0.9991927826	0.9985937720	0.9984012793	9.74525 E -04	0.097
(0.6000,0.6000)	0.9991004049	0.9988796763	0.9981806744	0.9975031224	1.59872 E -03	0.16
(0.7000,0.7000)	0.9987757500	0.9985170174	0.9977971450	0.9964064722	2.37218 E -03	0.24
(0.8000,0.8000)	0.9984012793	0.9981048482	0.9974581631	0.9951119854	3.29456 E -03	0.33
(0.9000,0.9000)	0.9979770489	0.9976432089	0.9972837348	0.9936204364	4.36544 E -03	0.44
(1.0000,1.0000)	0.9975031224	0.9971321487	0.9973810576	0.9919327166	5.58435 E -03	0.56
CPU time	NA	0.460000s	0.430000s	0.399600s		

Table 4: Numerical Comparison of Example 2 for $\Delta x = 0.1$, $\Delta t = 0.05$, $L = 10$

(x,t)	$u(x,t)$	UM	LM	Method (33)	Relative error	Percentage error (%)
(0.1000,0.1000)	0.9999900991	1.000150539	1.000294258	1.0005001250	5.10031 E -04	0.051
(0.2000,0.2000)	0.9999230799	1.000304064	1.000350801	0.9999000050	2.30767 E -05	0.0023
(0.3000,0.3000)	0.9997523243	1.000452525	1.000336322	0.9996000800	1.52282 E -04	0.015
(0.4000,0.4000)	0.9994484280	1.000569182	1.000136052	0.9991004049	3.48215 E -04	0.035
(0.5000,0.5000)	0.9990004998	1.001112699	0.9998403420	0.9984012793	5.99820 E -04	0.060
(0.6000,0.6000)	0.9984130253	0.9930245386	0.9993584603	0.9975031224	9.11349 E -04	0.091
(0.7000,0.7000)	0.9977006342	1.041218330	0.9987797082	0.9964064722	1.29714 E -03	0.13
(0.8000,0.8000)	0.9968829170	0.9006435880	0.9980268384	0.9951119854	1.77647 E -03	0.18
(0.9000,0.9000)	0.9959804757	1.111721077	0.9972063456	0.9936204364	2.36956 E -03	0.24
(1.0000,1.0000)	0.9950124792	0.9510926234	0.9963208653	0.9919327166	3.09520 E -03	0.31
CPU time	NA	0.510000s	0.370000s	0.250000s		

Table 5: Numerical Comparison of Example 3 for, $\Delta x = 0.1$, $\Delta t = 0.05$

(x,t)	$u(x,t)$	UM	LM	Method (33)	Relative error	Percentage error (%)
(0.1000,0.1000)	0.09090909091	0.0904875000	0.0904875000	0.09090909091	1.000000 E -16	0.00
(0.2000,0.2000)	0.1666666667	0.1653231744	0.1653231744	0.1666666667	1.000000 E -16	0.00
(0.3000,0.3000)	0.2307692308	0.2283156714	0.2283156715	0.2307692308	1.000000 E -16	0.00
(0.4000,0.4000)	0.2857142857	0.2821173696	0.2821192653	0.2857142857	1.000000 E -16	0.00
(0.5000,0.5000)	0.3333333333	0.3289352242	0.3286585068	0.3333333333	1.000000 E -16	0.00
(0.6000,0.6000)	0.3750000000	0.3543939265	0.3694124359	0.3750000000	1.000000 E -16	0.00
(0.7000,0.7000)	0.4117647059	0.6111136606	0.4055658801	0.4117647059	1.000000 E -16	0.00
(0.8000,0.8000)	0.4444444444	0.00861594345	0.4381714651	0.4444444444	1.000000 E -16	0.00
(0.9000,0.9000)	0.4736842105	38.46493835	0.4681540744	0.4736842105	1.000000 E -16	0.00
(1.0000,1.0000)	0.5000000000	0.1220675888	0.4964220517	0.5000000000	1.000000 E -16	0.00
CPU time	NA	0.350000s	0.320000s	0.281250s		

observed that the Method shows high computational strength as it compares favourably well with some known standard methods reported. [3]

From table 6 above, it could also be observed that the Method shows some superiority in-terms of accuracy when compared with the method of Wang *et al.* [11] and also compares favourably well with some known standard methods reported. [4]

7. Summary and Discussion of Results

This work contains a new class of multi-derivative method. The multi-derivative method was considered up to the second

---

[3] (x,t) is the coordinate of x and t, $u(x,t)$ is the exact solution, UM is the Upwind method, LM is the Lax method, Method represents the solution using the Method and Relative error=$| u(x,t) - $ Method $|/u(x,t)$. Percentage error = Relative error × 100%. CPU time is the method's computational time.

[4] (x,t) is the coordinate of x and t, $u(x,t)$ is the exact solution, UM is the Upwind method, LM is the Lax method, Method represents the solution using the Method and Absolute error=$| u(x,t) - $ Method $|$. CPU time is the method's computational time.

Table 6: Numerical Comparison for Example 4 for $v(x, t) = u(x, t)$, $\Delta x = 0.01$, $\Delta t = 0.01$, $\epsilon = 10^{-4}$

(x, t)	$u(x, t)$	UM	LM	Wang et al. [11]	Method (40)	Absolute error
(0.0000,0.0000)	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
(0.0100,0.0100)	0.01000100000	0.01009999010	0.01009999010	0.01000000000	0.01000010000	9.000000 E -07
(0.0300,0.0300)	0.03000300000	0.03089993849	0.03089993849	0.03000000000	0.03000030000	2.700000 E -06
(0.0500,0.0500)	0.05000500000	0.05249984036	0.08002383910	0.05000000000	0.05000050000	4.500000 E -06
(0.0600,0.0600)	0.06000600000	0.06359977227	0.1648091770	0.06000000000	0.06000060000	5.400000 E -06
(0.0800,0.0800)	0.08000800000	0.08639959684	0.4698987551	0.08000000000	0.08000080000	7.200000 E -06
(0.1000,0.1000)	0.1000100000	1.009904485	0.8868283796	0.1000000000	0.1000010000	4.500000 E -06
CPU time	NA	0.400000s	0.171875s	NA	0.078125s	

Table 7: Error Comparison for Example 1 for $v(x, t) = 1.0$.

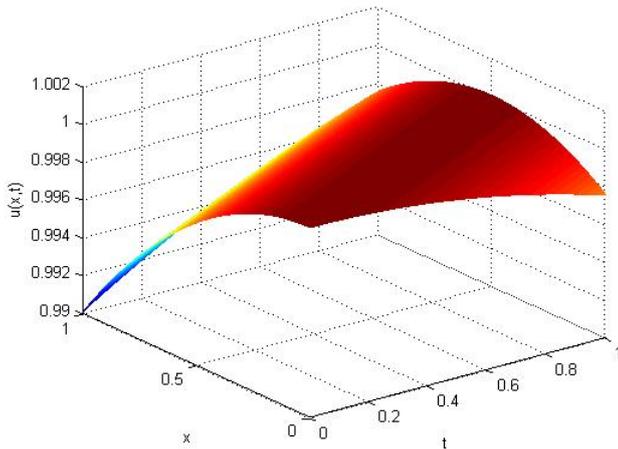| N	Adrain (2019) PC4 Scheme	Adrain (2019) PC4 Scheme	Proposed Method
40	6.43092 E -03	1.81277 E -04	5.58435 E -04
60	9.73768 E -04	2.78696 E -07	3.36544 E -09
80	2.71484 E -04	5.66109 E -09	3.24956 E -10
100	1.06781 E -04	2.87180 E -09	1.33524 E -10

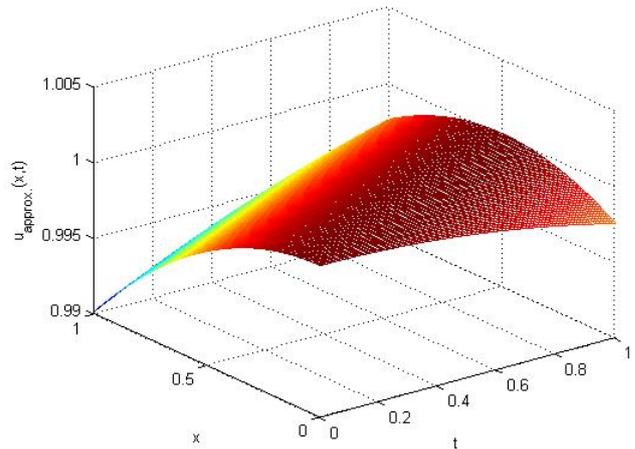Figure 4: Graph of Exact Solution of Example 1

Figure 5: Graph of Numerical Solution of Example 1

derivative ($k = 2$). The method was developed using some selected off-step points in its formation (this is to improve its accuracy), evaluation such as collocation and interpolation of these points were carried out and a system of algebraic equations was obtained for each value of k together with number of off-step points considered. These equations were solved to obtain the unknown parameters. The interpolating function is then evaluated at the selected collocation points to obtain a block form of this class of methods. These methods were then analyzed and were found to be stable and convergent.

Table 3-7 show the numerical computational results on problems advection equations. Extensive comparison was done with existing and standard methods as shown in the literature. These methods were demonstrated on some Examples and the superiority of the derived methods over these existing methods was established. It is worthy to say that the derived methods exhibit stronger computational strength. Also, Figures 4-11, in pairs,

show the graphical comparison of derived methods with exact for the advection problems considered.

8. Conclusion

Multi-derivative techniques are numerical tools designed for approximating singular problems. These methodologies use the evaluations of several derivatives at some prior points. Due to the characterized multi-step nature of some class of these methods, additional methods of evaluation called the predictor are always required in computation of previous values. This study produces a block form of these methodologies which need no predictions and thereby taking care of this drawback. Multi-derivative techniques have a long history of improvement in integrating Ordinary Differential Equations, but to date, a few class of these techniques have been explored in the integration of Partial Differential Equations. This research work explored
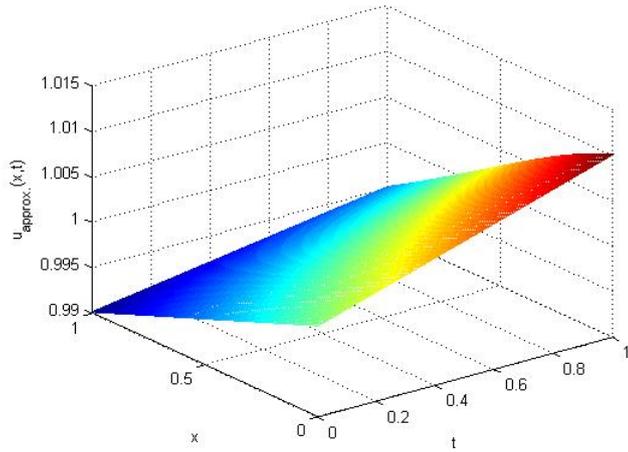
Figure 6: Graph of Exact Solution of Example 2
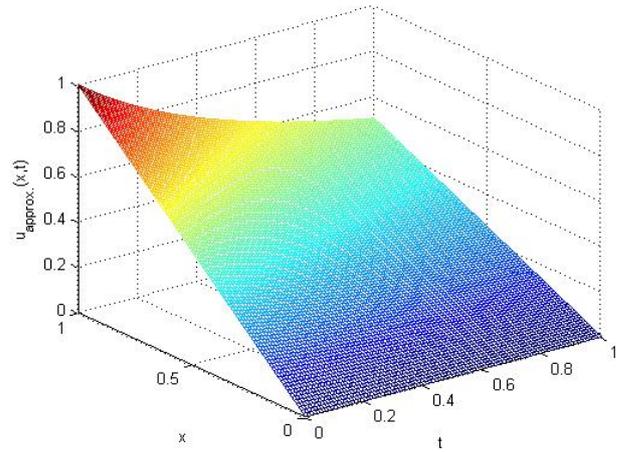

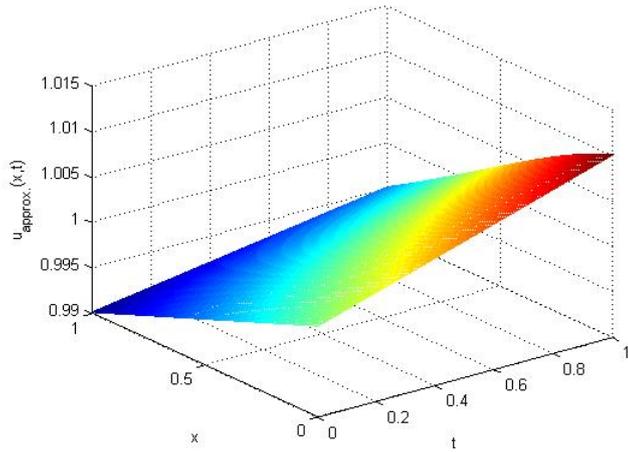Figure 9: Graph of Numerical Solution of Example 3

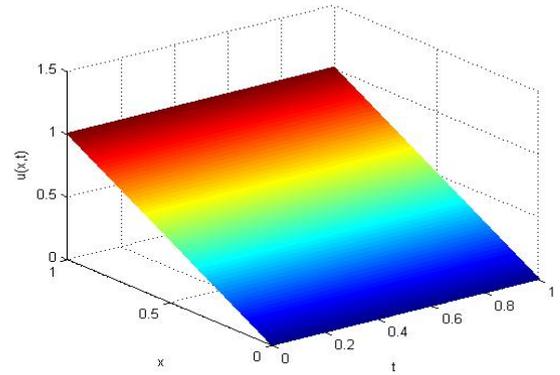Figure 7: Graph of Numerical Solution of Example 2

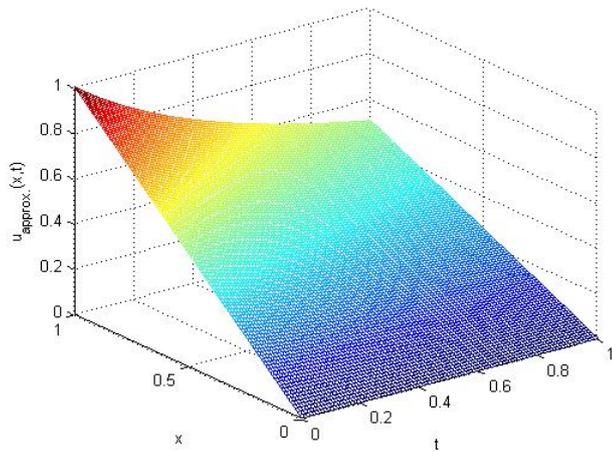Figure 10: Graph of Exact Solution of Example 4
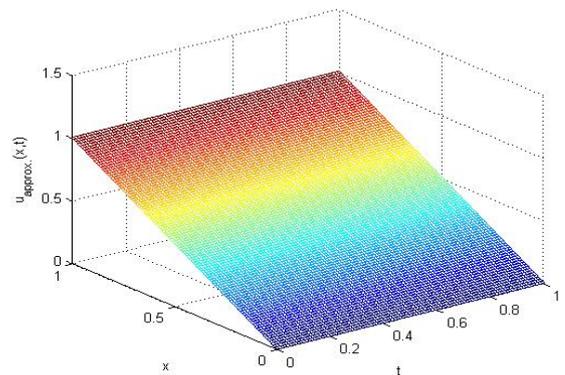

Figure 8: Graph of Exact Solution of Example 3

Figure 11: Graph of Numerical Solution of Example 4

the possibility of multi-derivative methods in integrating singular advection equations. From the computational results of this study, it could be observed that the derived methodologies solve satisfactorily the advection problems.

Acknowledgment

References

[1] A. Balzano, "Mosquito: An Efficient Finite Difference Scheme for Numerical Simulation of 2D Advection Equation", International Journal for Numerical Methods in Fluids **32** (1999) 2.

[2] B. J. Noye and M. Dehghan, "New explicit finite difference schemes for two-dimensional diffusion subject to specification of mass", Numerical Methods For Partial Differential Equations **15** (1999) 4.

[3] Y. Han, X. Zhang, L. Liu, Y. and Wu, "Multiple positive solutions of singular nonlinear Sturm-Liouville problems with Caratheodory perturbed term", Journal of Applied Mathematics, (2012) 160891.

[4] F. Z. Geng and S. P. Qian, "Solving singularly perturbed multipantograph delay equations based on the reproducing kernel method", Abstract and Applied Analysis, (2014) 794716.

[5] M. Stojanovic, "Splines difference methods for a singular perturbation problem", Applied Numerical Mathematics, **21** (1996) 3.

[6] S. Bushnaq, B. Maayah, S. Momani, and A. Alsaedi, "A Reproducing Kernel Hilbert Space Method for Solving Systems of Fractional Integro differential Equations", Abstract and Applied Analysis (2014) 103016.

[7] Z. Xianyl, "Linear Hybrid-variable Methods for Advection Equations", Advances in Computational Mathematics, Journal of Differential Equations, Conf **23** (2016) https://doi.org/10.1007/s10444-018-9647-z.

[8] E. Utku, S. Murat and K. Huseyin, "Efficient Numerical Treatment of Nonlinearities in the Advection-Diffusion-Reaction Equation", International Journal of Numerical Methods for Heat and Fluid Flow, http://doi.org/10.1108/HFF-02-2017-0198.

[9] Z. Gaetamo and P. Silvia, "On the Fractional Diffusion-Advection Equation for Fluids and Plasmas", Fluid **4** (2019) 62.

[10] S. Adrian, "A Generalization of Prefactored Compact Schemes for the Solution of Adevection Equations", arXiv:1902.04429v1 math.NA (2019).

[11] Y. Wang, H. Yu, G. Tan, and S. Qu, "Solving a Class of Singularly Perturbed Partial Differential Equations by Using the Perturbation Method and Reproducing Kernel Method", Hindawi Abstract and Applied Analysis (2014) 5.