Computational DIQM Scheme for Solving Nonlinear Volterra Integro-Differential Equations

Norhan A. Mohamed*, Sara I. Abo-Hashem, Salwa A. Mohamed

Department of Engineering Mathematics, Faculty of Engineering, Zagazig University, Egypt.

**ARTICLE INFO**

**Keywords**
- Volterra Integro-differential equation
- Differential quadrature method
- Integral quadrature method
- Exponential convergence rates

**ABSTRACT**

The authors in this work develop the differential quadrature method (DQM) to solve the nonlinear Volterra integro-differential equation by introducing an integration matrix operator that is fully combined with the differentiation matrix in (DQM) method. The obtained method (DIQM) transforms the discretized nonlinear Volterra integro-differential equation into a nonlinear algebraic system of equations which is solved iteratively by Newton's method. The efficiency of the (DIQM) method is examined by solving three examples where the error norms and convergence rates achieve the expected exponential behaviour.

**1. Introduction**

In recent years, there has been a clear interest in the integro-differential equations because some of the physical phenomena in engineering and physics fields cannot be described by differential equations only. In many cases, the effects of the past history of the system is needed to be reflected in the model which is described as a function of space and time and these effects of the previous history are ignored. Therefore, the way to overcome this problem is including an integral term in the differential equation to reflect all the preceding states through which the system has passed; that is to say, it is considered as a hereditary phenomenon which leads to integro-differential equations (IDEs) [1, 2].

Integro-differential equations play an important role to describe many processes like physical problems including thermoelastic contact [3], visco-elasticity [4-6], the theory of heat conduction [7-9], fluid dynamics, dropwise condensation and population [10–13]. Analytical solutions of integro-differential equations are hard to compute or they do not exist. Due to this, several numerical methods are developed to get an efficient numerical solution of integro-differential equation by many authors. In [14], an explicit integration method is used for solving a parabolic partial integro-differential equation (PIDE). The numerical solutions by finite element procedures are presented in [15-17]. Volterra equations are classified as IDEs that have been introduced by Vito Volterra (1860-1940). Aggarwal and Gupta [18] applied Kamal Transform for solving Linear Volterra Integro-differential equations of the second kind while in [19], they derived mixed interpolation methods for first and second-order VIDEs with periodic solutions. Also, Brunner et al. in [20] extend recent results for the numerical solution of Volterra integro-differential equations with periodic solution. The Haar wavelets are applied for solving of nonlinear Volterra integral equations and integro-differential equations in [21] and Legendre wavelet method is developed by Sahu and Ray [22] to approximate the solutions of the system of nonlinear Volterra integro-differential equations. Also, the nonlinear Volterra integro-differential equations with weakly singular kernels are solved by using spectral collocation methods as in [23, 24].

Rohaninasab and his co-workers presented a numerical solution of high-order Volterra integro-
differential equations by using Legendre collocation method in [25]. Many other numerical methods are used for solving these equations as Aguilar and Brunner in [26] who are using collocation methods to investigate the second-order VIDE in polynomial spline spaces. Chebyshev spectral methods are used to solve the nonlinear Volterra–Hammerstein integral equations [27]. In [28] a finite difference scheme is proposed for the weakly singular kernel of PIDEs. Hyperbolic PIDEs is discussed in [29] by spectral method. The numerical solution of the nonlinear parabolic Volterra PIDE is proposed in [30] by using the finite difference scheme for time and radial basis functions for space discretization. An analysis of Spectral collocation is proposed in [31] for the weakly singular kernel PIDEs. Also, several successive approximation methods for solving integro-differential equations are listed in literature such as Taylor polynomial [32], wavelet-Galerkin method [33], meshless method [34] and Taylor collocation method [35].

One of the most efficient numerical methods which appear in the last decades to solve differential equations is the differential quadrature method (DQM). It was first put by Bellman and Casti in the early 1970s [36], which is used for solving ordinary and partial differential equations by discretizing the derivatives at all the domains. Differential Quadrature Method (DQM) attracted the interest of many authors and because of that, its applications rapidly developed. Many researchers have done different types of DQMs using diverse base functions such as Hermit polynomials [37], spline functions [38], B-spline functions [39], Sinc functions [40].

The advantage of the differential quadrature method appears when it is used to solve boundary-value, initial-value, linear or nonlinear differential equations that DQM requires fewer grid points to obtain acceptable accuracy unlike finite difference method (FDM), finite element method (FEM) and finite volume method (FVM) which may need more number of grid points to obtain the solution. The authors in [41-43] succeeded to develop the DQM to solve problems that have an integral term in its differential equation by merging a new integrating matrix operator in the method and obtained differential integral quadrature method (DIQM). Also, by applying DIQM to other applications like nano-beams [44-47], it succeeds to obtain accurate solutions. Therefore, the strategy in this work reposes mainly on establishing an algorithm based on DIQM to obtain the solution of the Volterra integro-ordinary differential equations.

2. Problem Formulation

The authors consider the following nonlinear Volterra integro ordinary differential equation (VIODE):

\[
\frac{du}{dt} = f(t) + F(k) , \quad t \in [0,T] , \quad u(0) = a. \quad (1)
\]

where; \(a\) is constant value and \(F(k)\) is a Volterra integral operator which is defined by:

\[
F(k) = \int_0^t K(t,s) g(u(s)) \, ds . \quad (2)
\]

The source function \(f\), the nonlinear function \(g\) and kernel function \(K\) are assumed to be sufficiently smooth. This integro-differential equation is called a second kind Volterra integral equation since the unknown function \(u\) appears both inside and outside the integral sign of the equation.

3. The Solution Procedure

3.1. DQM Method

Let the domain \(0 \leq t \leq T\) of a problem be discretized using \(N\) nodes; \(t_1 = 0, t_2, \ldots, t_N = T\). Using the DQM, different order derivatives of a function at a given node can be approximated using a weighted sum of the function values at all discrete nodes in that domain. For example, the first derivative of function \(f(t)\) at node \(t_i\) can be approximated using the DQM as follows:

\[
\frac{df}{dt} \bigg|_{t=t_i} \approx \sum_{j=1}^{N} a_{ij} f_j , \quad i = 1,2,\ldots,N , \quad (3)
\]

where \(f_j = f(t_j)\) and \(a_{ij}\) denote the corresponding weighting coefficients. The weighting coefficients for the first derivative can be expressed as follows, Shu [48]:

\[
a_{ij} = \frac{1}{t_j-t_i} \left( \frac{p_i}{p_j} \right) , \quad i \neq j \quad \text{and} \quad a_{ii} = -\sum_{j=1, j \neq i}^{N} a_{ij}^2 , \quad (4)
\]

where

\[
P_i = \prod_{j=1, j \neq i}^{N} (t_i - t_j) , \quad i, j = 1,2,\ldots,N . \quad (5)
\]

In matrix form, the vector \(f = [f_1, f_2, \ldots, f_N]^T\) represents the discrete values \(f_i = f(t_i)\) at nodes \(t_1 = 0, t_2, \ldots, t_N = T\). According to Eq. (3), the DQM approximation of the first derivative is given by

\[
F = A f , \quad (6)
\]

where \(F = [f_1', f_2', \ldots, f_N']^T, \quad f_i' = \left( \frac{df}{dt} \right)_{t_i} \) and
\( A = [a_{ij}] \) is the weighting \((N \times N)\) matrix of the first order derivative that can be easily computed using Eqs. (4, 5). The weighting coefficients matrices for higher-order derivatives can be determined via matrix multiplication.

For the purpose of efficiency, the discrete nodes on the domain are distributed according to the Chebyshev–Gauss–Lobatto formula

\[
t_i = \frac{T}{2} \left(1 - \cos \frac{i-1}{N-1} \pi \right), \quad i = 1, 2, \ldots, N.
\]

3.2 The Integral Quadrature Method (IQM)

Consider a differentiable function \( f(t) \) defined on the interval \( a \leq t \leq b \). In continuous calculus, if

\[
\frac{df(t)}{dt} = F(t),
\]

Integrating both sides of Eq.(8), then \( \int F(t) = f(t) + c \) and \( f(t) \) is called an antiderivative of \( F(t) \). In this work, we are interested in the definite integral

\[
\int_a^b F(t)dt = (f(b) + c) - (f(a) + c) = f(b) - f(a).
\]

It is noticed that its value is independent the arbitrary constant \( c \). Returning to DQM, Eq. (6) is discretized in matrix form as \( A \hat{f} = \hat{F} \) (see Eq. (6)). To inverse this differentiation process, one can write \( \hat{f} = A'^{-1} \hat{F} \). However matrix \( A' \) is singular and hence has no inverse. This is expected and interpreted mathematically since the anti-derivative of a given function is not unique due to the presence integration constant. The main idea behind defining the definite integral matrix operator is to replace the nonexistent \( A^{-1} \) by a pseudo-inverse matrix \( A'^+ \) of \( A \). Mohamed [49] has proven that \( A'^+ \) acts as anti-derivative operator with respect to the differential operator \( A \) such that

\[
A'^+ F = f + C,
\]

where \( C \) is a vector with all of its component equals \( c = \frac{1}{N} \sum_{i=1}^{N} f_i \). That is \( A'^+ \) reverses the derivative operator \( A \) but for an additive constant. Note that, in Eq. (10), the \( i \)th component can be written as

\[
(A'^+ F)_i = \sum_{k=1}^{N} A'^+ F_k = f_i + c.
\]

Consequently the definite integral between two nodes \( t_i, t_j \) is approximated as

\[
\int_{t_i}^{t_j} F(t)dt = f(t_j) - f(t_i) \equiv \sum_{k=1}^{N} (A'^+ F_k - A'^+ F_k) F_k,
\]

where ‘\( \equiv \)’ stands for ‘is discretized and approximated by’.

Note that the row vector \( R_{ij} = (A'^+ j_k - A'^+ i_k), k = 1, 2, \ldots, N \) is just the difference between the \( j \)th and \( i \)th rows of matrix \( A'^+ \). Eq. (12) is written in matrix form as

\[
\int_{t_i}^{t_j} F(t)dt \equiv R_{ij} \hat{F}.
\]

Next we consider approximation of \( \int_0^T F(t)dt \), \( 0 < t < T \). Define the integral matrix \( R \) as

\[
R = A'^+ - \hat{A}^+,
\]

where \( \hat{A}^+ \) is an \( N \times N \) square matrix formed such that each of its rows equals the first row in \( A'^+ \). The definite integral operator \( R \) is interpreted through the following expanded form

\[
\left[ \int_0^{t_1} F(t)dt \right] \left[ \int_0^{t_2} F(t)dt \right] \left[ \int_0^{t_N} F(t)dt \right] \equiv \left[ \begin{array}{c} F_1 \\ R_{21} F_2 \\ \vdots \\ R_{N1} F_N \end{array} \right],
\]

where element \( R_{ij} = A'^+ i_j - A'^+ j_i \). The definite integral operator \( R \) has the following properties

\[
\int_0^T F(t)dt \equiv RF,
\]

\[
\int_0^{t_i} F(t)dt \equiv \hat{R}_i F.
\]

3.3. DIQM for Integro-Differential Equations

Consider the nonlinear Volterra integro ordinary differential equation Eqs.(1,2) with the independent variable \( 0 < t < T \). Using the Chebyshev–Gauss–Lobatto distribution Eq. (7), the domain is discretized by \( N \)-points. Let the discrete vectors \( \bold{t}, \bold{u}, \bold{f} \) for \( t \), unknown function \( u(t) \), and known function \( f(t) \), be respectively

\[
t = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix}, \quad f = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}.
\]

Let the \( N \times N \) matrices \( A \) and \( R \) be respectively the first order differential operator and the definite integral operator. First, consider the integral term...
\[ \mathbb{I}(t) = \int_0^t \mathcal{K}(t, s) g(u(s)) \, ds. \quad (19) \]

Note that the i-th component of \( \mathbb{I}(t) \) is \( I_i = \int_0^t \mathcal{K}(t, s) g(u(s)) \, ds \), its DIQM discretization is

\[ I_i \approx \sum_{j=1}^{N} R_{ij} \mathcal{K}(t_i, t_j) g(u_{ij}) \quad (20) \]

Accordingly, \( \mathbb{I}(t) \) can be discretized by DIQM as

\[ \mathbb{I}(t) \approx \sum_{j=1}^{N} R_{ij} \mathcal{K}(t_i, t_j) g(u_{ij}). \quad (21) \]

It must be mentioned that if \( \mathbb{I}(t) \) is a part in an integro-differential equation then its discrete algebraic system is nonlinear. To solve the resulting nonlinear system by Newton method one has to compute the Jacobian matrix for the system. To compute the Jacobian matrix corresponding to the integral term defined in Eq. (19), we first compute its general element by differentiating the right hand size of Eq. (20) to get

\[ J_{ij} = \frac{\partial I_i}{\partial u_j} = R_{ij} \mathcal{K}(t_i, t_j) g'(u_j). \quad (22) \]

In matrix form, the Jacobian matrix for the right hand side of Eq. (20) is obtained as

\[ J = R \circ \mathcal{K}(T, T^T) \circ g'(U), \quad (23) \]

where

\[ U = \begin{bmatrix} u_1 & u_2 & \cdots & u_K \noalign{\medskip} u_1 & u_2 & \cdots & u_K \noalign{\medskip} \vdots & \vdots & \ddots & \vdots \noalign{\medskip} u_1 & u_2 & \cdots & u_K \end{bmatrix}, \quad g = \frac{dg}{dU}. \]

\[ T = \begin{bmatrix} t_1 & t_2 & \cdots & t_1 \noalign{\medskip} t_2 & t_2 & \cdots & t_2 \noalign{\medskip} \vdots & \vdots & \ddots & \vdots \noalign{\medskip} t_K & t_K & \cdots & t_K \end{bmatrix}. \quad (24) \]

Hadamard matrix operator ‘\( \circ \)’ is the element by element operator defined for matrices \( A, B, C \) having the same dimensions such that \( C = A \circ B \) implies that \( C_{ij} = A_{ij} B_{ij} \). In Eq. (20), \( \mathcal{K}(T, T^T) \) is a square \( N \)-matrix with general element \( \mathcal{K}_{ij} = \mathcal{K}(T_{ij}, T_{ij}^T) \).

This approach enables DIQM to transform integro-differential equation Eq. (1) into a system of nonlinear algebraic equations.

\[ F = Au - f - B g(u) = 0. \quad (25) \]

With the Jacobian matrix

\[ J = A - B \circ g'(U), \quad (26) \]

where matrix \( B = R \circ \mathcal{K}(T, T^T) \).

### 4. Numerical Results

In this section, we prove the effectiveness of our approach which is based on DIQM to obtain the approximate solution of nonlinear Volterra IODEs. All of the numerical computations are performed using MATLAB R2018b and the accuracy is measured for the proposed method by using the following two error norms definitions:

\[ L_2 = \sqrt{\sum_{i=1}^{N} (u_{\text{computed}}(i) - u_{\text{exact}}(i))^2}, \]

\[ L_\infty = \max_i |u_{\text{computed}}(i) - u_{\text{exact}}(i)|. \quad (27) \]

where, \( N \) is the number of discretized points. Also, the authors provide the convergence rate for all numerical examples where the numerical error is computed by the exponential order \( O(C^N) \) for \( 0 < C < 1 \). The convergence rate is faster when the value of \( C \) is smaller and this is achieved if the solution is smooth [50]. The convergence parameter \( C \) can be evaluated by:

\[ C = e^{-\frac{\ln(Err_1/Err_2)}{(N_1-N_2)}}, \quad (28) \]

where, \( Err_1 \) and \( Err_2 \) are the error norms computed at the domain which is discretized by \( N_1 \) and \( N_2 \) of grid points, respectively.

**Example 1**

We consider the following nonlinear VIODE [51]:

\[ \frac{du}{dt} = f(t) + \int_0^t s e^{-u^2(s)} \, ds, \quad t \in [0,1] \quad (29) \]

where \( f(t) = 1 - \frac{t}{2} + \frac{t^3}{2} e^{-t^2} \) with initial condition \( u(0) = 0 \) and exact solution \( u(t) = t \).
Applying the proposed DIQM, Eq. (29) is discretized into a nonlinear algebraic system in the form of Eq. (25) where

$$g(u) = \begin{bmatrix} e^{-u_1^2}, e^{-u_2^2}, \ldots, e^{-u_K^2} \end{bmatrix}^T, \quad B = R \circ \mathcal{K}(T, T^T).$$

In this example, $\mathcal{K}(t,s) = t \cdot s$, then

$$\mathcal{K}(T, T^T) = T \circ T^T = \begin{bmatrix} t_1 t_1 & t_1 t_2 & \cdots & t_1 t_K \\ t_2 t_1 & t_2 t_2 & \cdots & t_2 t_K \\ \vdots & \vdots & \ddots & \vdots \\ t_K t_1 & t_K t_2 & \cdots & t_K t_K \end{bmatrix}.$$

The Jacobian for this algebraic system is given by Eq. (26). Note that since $g'(u) = -2u e^{-u^2}$, then

$$g'(U) = -2 \begin{bmatrix} u_1 e^{-u_1^2} & u_2 e^{-u_2^2} & \cdots & u_K e^{-u_K^2} \\ u_1 e^{-u_1^2} & u_2 e^{-u_2^2} & \cdots & u_K e^{-u_K^2} \\ \vdots & \vdots & \ddots & \vdots \\ u_1 e^{-u_1^2} & u_2 e^{-u_2^2} & \cdots & u_K e^{-u_K^2} \end{bmatrix}.$$

Table 1: $L_2$ and $L_{\infty}$ error norms and the convergence rates for Example 1 at $t=1$.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$M$</th>
<th>$L_2$</th>
<th>$C$</th>
<th>$L_{\infty}$</th>
<th>$C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>4.49E-04</td>
<td>-----</td>
<td>3.67E-04</td>
<td>-----</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>3.46E-06</td>
<td>0.2963</td>
<td>2.62E-06</td>
<td>0.2907</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>8.39E-08</td>
<td>0.3947</td>
<td>5.16E-08</td>
<td>0.3746</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>1.20E-09</td>
<td>0.3459</td>
<td>7.02E-10</td>
<td>0.3415</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>1.34E-11</td>
<td>0.3251</td>
<td>7.25E-12</td>
<td>0.3188</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>1.22E-13</td>
<td>0.3090</td>
<td>6.11E-14</td>
<td>0.3029</td>
</tr>
<tr>
<td>16</td>
<td>5</td>
<td>8.71E-16</td>
<td>0.2905</td>
<td>4.44E-16</td>
<td>0.2920</td>
</tr>
</tbody>
</table>

The convergence results are presented in Table 1. The $L_2$ and $L_{\infty}$ error norms are reported for different numbers of grid points $N$. The number $M$ of iteration steps in Newton method required to reduce the error norms below $\varepsilon = 10^{-15}$ is also reported. To demonstrate the spectral convergence rate, the convergence parameter $C$ is computed according to Eq. (28) and presented in Table 1 showing that $0.29 < C < 0.4 < 1$. The results illustrate that accurate results are obtained using few grid points and that $N=16$ is sufficient to reduce the error below the computer round off error.

These results are also presented in Fig. 1 on a semi-log scale where the error logarithm is plotted versus the number of grid points. For both of the $L_2$ and $L_{\infty}$ error norms, the nearly straight line plot indicates that the errors decay exponentially. The $L_2$ error norms computed at different times in the interval $0.2 < t < 1$ are plotted in Fig. 2 for $N=4, 8, 16$. It is observed that although the error increases with time, the solution is still accurate in this range of time.

**Example 2**

Consider the following nonlinear VIODE [51]:

$$\frac{du}{dt} = f(t) + \int_0^t (t - s) \ln(1 + u(s)) \, ds, \quad t \in [0,1] \quad (30)$$

where,

$$f(t) = 1/24(8 + 9t^2 + \frac{12}{\sqrt{\pi}} - 8\sqrt{1 + \tau} - 4t(-6 + 5\sqrt{1 + \tau} - 12t^2 \ln(1 + \sqrt{1 + \tau})).$$

with the initial condition $u(0)=1$, and the exact solution is given by $u(t) = \sqrt{1 + \tau}$.

---

Fig. 1: Error norms for different values of $N$ at $t=1$ of Example 1.

Fig. 2: $L_2$ for different values of time and $N$ for Example 1.
By following the same steps in example (1), we can get the Jacobian matrix of this example by Eq. (26) where:

\[
g'(U) = \begin{bmatrix}
\frac{1}{1+u_1} & \frac{1}{1+u_2} & \cdots & \frac{1}{1+u_K} \\
\frac{1}{1+u_1} & \frac{1}{1+u_2} & \cdots & \frac{1}{1+u_K} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{1}{1+u_1} & \frac{1}{1+u_2} & \cdots & \frac{1}{1+u_K}
\end{bmatrix}.
\]

In Table 2, \(L_2\) and \(L_\infty\) error norms are listed for different numbers of grid points \(N\). Also, the convergence rate results are presented in Table 2 where its range is \(0.34 < C < 0.4 < 1\). This example needs \(M = 10\), number of iteration steps in Newton method, to reduce the error norms to \(\varepsilon = 10^{-13}\). We need only grid points \(N = 16\) to obtain accurate results.

Table 2: \(L_2\) and \(L_\infty\) error norms and the convergence rate for Example 2 at \(t = 1\).

<table>
<thead>
<tr>
<th>(N)</th>
<th>(M)</th>
<th>(\text{L}_2)</th>
<th>(C)</th>
<th>(\text{L}_\infty)</th>
<th>(C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>10</td>
<td>1.29E-03</td>
<td></td>
<td>8.09E-04</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>1.80E-05</td>
<td>0.3441</td>
<td>1.17E-05</td>
<td>0.3472</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>3.62E-07</td>
<td>0.3763</td>
<td>2.15E-07</td>
<td>0.3677</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>8.13E-09</td>
<td>0.3872</td>
<td>4.39E-09</td>
<td>0.3782</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>1.94E-10</td>
<td>0.3930</td>
<td>9.64E-11</td>
<td>0.3849</td>
</tr>
<tr>
<td>14</td>
<td>10</td>
<td>4.81E-12</td>
<td>0.3968</td>
<td>2.22E-12</td>
<td>0.3895</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>1.23E-13</td>
<td>0.4002</td>
<td>5.31E-14</td>
<td>0.3932</td>
</tr>
</tbody>
</table>

Fig. 3: Error norms for different values of \(N\) at \(t = 1\) of Example 2.

The \(L_2\) and \(L_\infty\) error norms are plotted in Fig. 3, the error logarithm on a semi-log scale versus the number of grid points, and the results indicate that the errors decay exponentially. In Fig. 4, the \(L_2\) error norms are plotted for \(N = 4, 8, 16\) which computed in the interval \(0.2 < t < 1\). As in example 1, the accuracy is still achieved even with an increase of errors.

**Example 3**

We investigate the following nonlinear VIODE [52]:

\[
\frac{du}{dt} = f(t) + \int_0^t e^{-s} u^3(s) \, ds, \quad t \in [0,1]
\]

(31)

where, \(f(t) = \frac{3}{2} e^t - \frac{1}{2} e^{3t}\) with initial condition \(u(0) = 1\), and the exact solution is given by \(u(t) = e^t\).

Also, the Jacobian matrix in this example is obtained as we can get it in the previous examples. And \(g'(u) = 3u^2\), then:

\[
g'(U) = 3 \begin{bmatrix}
u_1^2 & u_2^2 & \cdots & u_K^2 \\
u_1^2 & u_2^2 & \cdots & u_K^2 \\
\vdots & \vdots & \ddots & \vdots \\
u_1^2 & u_2^2 & \cdots & u_K^2
\end{bmatrix}.
\]

Table 3: \(L_2\) and \(L_\infty\) error norms and the convergence rate for Example 3 at \(t = 1\).

<table>
<thead>
<tr>
<th>(N)</th>
<th>(M)</th>
<th>(\text{L}_2)</th>
<th>(C)</th>
<th>(\text{L}_\infty)</th>
<th>(C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>10</td>
<td>1.85E-01</td>
<td></td>
<td>1.72E-01</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>2.71E-04</td>
<td>0.1957</td>
<td>2.23E-04</td>
<td>0.1897</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>8.85E-07</td>
<td>0.2390</td>
<td>6.49E-07</td>
<td>0.2321</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>2.12E-09</td>
<td>0.2213</td>
<td>1.41E-09</td>
<td>0.2161</td>
</tr>
</tbody>
</table>
As in the previous examples, Table 3 presented that the convergence rate results range from $0.19 < C < 0.96 < 1$ of this example. The $L_2$ and $L_\infty$ error norms need the numbers of grid points $N=16$ to obtain the error norms of $\varepsilon = 10^{-15}$. Figs. 5 shows that the $L_2$ and $L_\infty$ error norms are still decay exponentially and the accuracy is still achieved although the error increases with time as shown in Fig. 6.

5. Conclusion

In this paper, the authors presented the differential integral quadrature method (DIQM) by combining an integration matrix operator with the differentiation matrix in (DQM) method. The obtained method (DIQM) is applied for solving the nonlinear Volterra integro-differential equation (VIDE). The results of the three provided examples illustrate that the proposed method is efficient and achieve the expected exponential behavior of accuracy by using a few numbers of grid points. In future work, the proposed method (DIQM) is easy to implement for solving other kinds of integro-differential equations.

References


