Research Article

Characteristics of the Differential Quadrature Method and Its Improvement

Wang Fangzong, Liao Xiaobing, and Xie Xiong

College of Electrical Engineering & New Energy, China Three Gorges University, Yichang, Hubei 443002, China

Correspondence should be addressed to Wang Fangzong; fzwang@ctgu.edu.cn

Received 8 September 2014; Revised 23 December 2014; Accepted 25 December 2014

Academic Editor: Ming-Hung Hsu

Copyright © 2015 Wang Fangzong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The differential quadrature method has been widely used in scientific and engineering computation. However, for the basic characteristics of time domain differential quadrature method, such as numerical stability and calculation accuracy or order, it is still lack of systematic analysis conclusions. In this paper, according to the principle of differential quadrature method, it has been derived and proved that the weighting coefficients matrix of differential quadrature method meets the important V-transformation feature. Through the equivalence of the differential quadrature method and the implicit Runge-Kutta method, it has been proved that the differential quadrature method is A-stable and s-stage s-order method. On this basis, in order to further improve the accuracy of the time domain differential quadrature method, a class of improved differential quadrature method of s-stage 2s-order have been proposed by using undetermined coefficients method and Padé approximations. The numerical results show that the improved differential quadrature method is more precise than the traditional differential quadrature method.

1. Introduction

The differential quadrature method (DQM) was first proposed by Bellman and his associates in the early 1970s [1, 2], which is usually used for solving ordinary and partial differential equations. As an analogous extension of the quadrature for integrals, it can be essentially expressed as the values of the derivatives at each grid point as weighted linear sums approximately of the function values at all grid points within the domain under consideration.

The differential quadrature method is conceptually simple and the implementation is straightforward. It has been recognized that the differential quadrature method has the capability of producing highly accurate solutions with minimal computational effort [3, 4] when the method is applied to problems with globally smooth solutions. So far, the differential quadrature method has been widely applied to boundary-value problems in many areas of engineering and science, such as structural mechanics [5–8], transport process [9], dynamic systems [10–12], and calculation of transmission line transient response [13, 14]. A comprehensive review of the chronological development of the differential quadrature method can be found in [4]. Although the differential quadrature method has been successfully applied in so many fields, for the basic characteristics of the method, such as numerical stability and calculation accuracy or order, not much work about them has been done in this area for the differential quadrature method. According to Fung [15], using Lagrange interpolation functions as test functions, the differential quadrature in time domain was shown to be equivalent to the recast implicit Runge-Kutta method [16–18]; besides, some low-order algorithms were discussed in detail. However, the method used by Fung is not the traditional sense of differential quadrature method but involved postprocessing (i.e., numerical solution at the end of grid points adopts polynomial extrapolation).

In this paper, using general polynomial as test functions [19], the weighting coefficients matrix of differential quadrature method is proved to satisfy V-transformation [17, 20]. The equivalent implicit Runge-Kutta method is constructed through the differential quadrature method. Hence, making use of Butcher fundamental order theorem and the method of linear stability analysis [17, 18], the basic characteristics of the differential quadrature method can be systematically
analysed. Unfortunately, the differential quadrature method is only a method of s-stage s-order and A-stable. Consequently, the differential quadrature method cannot yield higher accurate solutions to the boundary-value problems with fewer computational efforts. Based on above deduction, the method of undetermined coefficients is used to make the stability function of the equivalent Runge-Kutta method become the diagonal Padé approximations to the exponential function [17, 18]. Therefore, a class of improved differential quadrature method of s-stage 2s-order is derived.

The paper is arranged as follows. In Section 2, the weighting coefficients matrix of traditional differential quadrature method using general polynomial as test functions is briefly discussed. In Section 3, the equivalent relationship between the differential quadrature method and the Runge-Kutta methods is deduced. In Section 4, the stability and accuracy characteristics of the differential quadrature method are studied. A class of improved differential quadrature method using general polynomial as test functions is briefly introduced. In Section 5, the first order and A-stable is derived. In Section 6, the transient response of a double-degree-of-freedom system is computed, which is given to verify the computational accuracy with the defined three grid points. Conclusions are then given in Section 7.

2. Traditional Differential Quadrature Method

Suppose function \( f(x) \) is sufficiently smooth in the whole interval; there are \((s + 1)\) grid points with coordinates as \( c_i, i \in (0, s)\). The first order derivative \( f^{(1)}(c_i) \) at each grid point \( c_i, i \in (1, s)\), is approximated by a linear sum of all the function values in the whole domain; that is,

\[
 f^{(1)}(c_i) = \sum_{j=0}^{s} g_{ij} f(c_j), \quad i \in (1, s), \tag{1}
\]

where \( f(c_j) \) represent function values at a grid point \( c_j \), and \( g_{ij} \) is the weighting coefficients.

In order to compute the weighting coefficients \( g_{ij} \) in (1), the test functions can be chosen as

\[
 r_k(x) = x^k, \quad (k = 0, 1, \ldots, s). \tag{2}
\]

Substituting (2) into (1) gives

\[
 k = 0, \quad 0 = \sum_{j=0}^{s} g_{ij} r_j, \quad i \in (1, s), \tag{3}
\]

\[
 \sum_{j=0}^{s} g_{ij} c_j^k = k \cdot c_i^{k-1}, \quad i, k \in (1, s). \tag{4}
\]

Equation (3) can be expanded into matrix form as

\[
 \begin{pmatrix}
  g_{11} & g_{12} & \cdots & g_{1s} \\
  g_{21} & g_{22} & \cdots & g_{2s} \\
  \vdots & \vdots & \ddots & \vdots \\
  g_{s1} & g_{s2} & \cdots & g_{ss}
 \end{pmatrix}
 \begin{pmatrix}
  1 \\
  1 \\
 \vdots \\
  1
 \end{pmatrix}
 +
 \begin{pmatrix}
  g_{10} \\
  g_{20} \\
 \vdots \\
  g_{s0}
 \end{pmatrix}
 =
 \begin{pmatrix}
  0 \\
  0 \\
 \vdots \\
  0
 \end{pmatrix}. \tag{5}
\]

Let

\[
 G_0 = \begin{pmatrix} g_{10} \\ g_{20} \\ \vdots \\ g_{s0} \end{pmatrix}, \quad G = \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1s} \\ g_{21} & g_{22} & \cdots & g_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ g_{s1} & g_{s2} & \cdots & g_{ss} \end{pmatrix}, \quad e = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}. \tag{6}
\]

Using (6), (5) can be simplified to

\[
 G_0 \equiv -Ge. \tag{7}
\]

From \( k = 1, 2, \ldots, s \), (4) can be expanded as

\[
 g_{10} c_0^k + g_{11} c_1 + g_{12} c_2 + \cdots + g_{1s} c_s = 1
\]

\[
 g_{20} c_0^k + g_{21} c_1^2 + g_{22} c_2^2 + \cdots + g_{2s} c_s^2 = 2c_1
\]

\[
 \vdots
\]

\[
 g_{s0} c_0^k + g_{s1} c_1^s + g_{s2} c_2^s + \cdots + g_{ss} c_s^s = sc_s^{s-1}.
\]

Since initial grid point \( c_0 \) is usually defined as 0, (8) reduces to

\[
 g_{11} c_1 + g_{12} c_2 + \cdots + g_{1s} c_s = 1
\]

\[
 g_{11} c_1^2 + g_{12} c_2^2 + \cdots + g_{1s} c_s^2 = 2c_1
\]

\[
 \vdots
\]

\[
 g_{11} c_1^s + g_{12} c_2^s + \cdots + g_{1s} c_s^s = sc_s^{s-1}.
\]

From \( i = 1, 2, \ldots, s \), (9) can be also expanded into matrix form as

\[
 G = \begin{pmatrix}
  c_1 & c_2^2 & \cdots & c_s^2 \\
  c_2 & c_3^2 & \cdots & c_s^2 \\
 \vdots & \vdots & \ddots & \vdots \\
  c_s & c_s^2 & \cdots & c_s^2
 \end{pmatrix}
 = \begin{pmatrix}
  1 & 2c_1 & \cdots & sc_s^{s-1} \\
  1 & 2c_2 & \cdots & sc_s^{s-1} \\
 \vdots & \vdots & \ddots & \vdots \\
  1 & 2c_s & \cdots & sc_s^{s-1}
 \end{pmatrix}, \tag{10}
\]

Vandermonde matrix \( V \) is defined as follows:

\[
 V = \begin{pmatrix}
  1 & c_1 & \cdots & c_1^{s-1} \\
  1 & c_2 & \cdots & c_2^{s-1} \\
 \vdots & \vdots & \ddots & \vdots \\
  1 & c_s & \cdots & c_s^{s-1}
 \end{pmatrix}. \tag{11}
\]
Making use of (II), (10) can be expressed as
\[
G^{-1}V = \begin{pmatrix}
c_1 & c_2 & \cdots & c_s \\
c_2 & c_2 & \cdots & c_s \\
\vdots & \vdots & \ddots & \vdots \\
c_s & c_s & \cdots & c_s
\end{pmatrix}
\begin{pmatrix}
1 & \frac{1}{2} & \cdots & \frac{1}{s} \\
\frac{1}{2} & \frac{1}{2} & \cdots & \frac{1}{s} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{1}{s} & \frac{1}{s} & \cdots & \frac{1}{s}
\end{pmatrix}
\]
\[(12)\]

Finally, it can be inferred that
\[
V^{-1}G^{-1}V = A_s,
\]
where \(A_s\) is
\[
A_s = \begin{pmatrix}
0 & 0 & \cdots & \alpha_1 \\
1 & 0 & \cdots & \alpha_2 \\
0 & \frac{1}{2} & \cdots & \vdots \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \frac{1}{s-1} \alpha_s
\end{pmatrix},
\]
\[(13)\]
with
\[
\alpha_i = [\alpha_1, \alpha_2, \ldots, \alpha_s]^T = \frac{1}{s}V^{-1}c',
\]
\[
c' = [c_1', c_2', \ldots, c_s']^T.
\]
Equation (13), that is, \(G = VA_sV^{-1}\), is called the implicit expression of the weighting coefficients matrix of the differential quadrature method and is also called \(V\)-transformation.

When the grid points have been selected, the weighting coefficients matrices \(G\) and \(G_0\) are easy to calculate with the above formula. Obviously, the weighting coefficients of the differential quadrature method depend on the test functions and distribution of grid points but are independent of some specific problems. There are four typical grid points' distributions: Legendre grid points, Chebyshev grid points, Chebyshev-Gauss-Lobatto grid points, and Uniform grid points (also called equally spaced grid points) [18]. This paper will focus on the latter three kinds of commonly used grid points, which are defined as follows:

(1) Chebyshev grid points:
\[
c_k = \frac{1}{2} \left(1 - \cos \left(\frac{2k - 1}{2s - 2} \pi\right)\right), \quad k \in (1, s - 1),
\]
\[
c_0 = 0, \quad c_s = 1;
\]
(2) Chebyshev-Gauss-Lobatto grid points:
\[
c_k = \frac{1}{2} \left(1 - \cos \left(\frac{k}{s} \pi\right)\right), \quad k \in (0, s);
\]
(3) Uniform grid points:
\[
c_k = \frac{k}{s}, \quad k \in (0, s).
\]

\[(14)\]

3. The Equivalence of Differential Quadrature Method and Runge-Kutta Method

In order to analyse the numerical stability and order of the differential quadrature method, the differential quadrature method in time domain can be transformed into equivalent implicit Runge-Kutta method. Consider the following ordinary differential equation
\[
\frac{dx}{dt} = f(t,x), \quad 0 < t \leq T
\]
\[(15)\]
in the following, \(t_n, t_{n+1}\) represent, respectively, the beginning and the end points at each step. \(h = t_{n+1} - t_n\) will be used to denote the step size. The time interval \([t_n, t_{n+1}]\) will be normalized. That is, \(c = (t - t_n)/h, \ t \in [t_n, t_{n+1}], \ c \in [0,1]\). At the same time, (19) can be made in the standard normalized form
\[
\frac{dx}{dt} = hf(t_n + ch, x), \quad \bar{x} = x(t_n + ch);
\]
\[(16)\]
then, using \(s\)-stage differential quadrature method to solve (20) yields
\[
G \begin{pmatrix}
\bar{x}_1 \\
\vdots \\
\bar{x}_s
\end{pmatrix} + G_0x_n = h \begin{pmatrix}
f(t_n + c_1h, \bar{x}_1) \\
\vdots \\
f(t_n + c_sh, \bar{x}_s)
\end{pmatrix},
\]
\[(21)\]
where \(\bar{x}_i = x(t_n + ch), \ i \in (1, s).\) Since \(G_0 \equiv -Ge\), (21) reduces to
\[
\begin{pmatrix}
\bar{x}_1 \\
\vdots \\
\bar{x}_s
\end{pmatrix} = x_n \begin{pmatrix}
1 \\
\vdots \\
1
\end{pmatrix} + hG^{-1} \begin{pmatrix}
f(t_n + c_1h, \bar{x}_1) \\
\vdots \\
f(t_n + c_sh, \bar{x}_s)
\end{pmatrix}.
\]
\[(22)\]
Let
\[
G^{-1} = A_s = \begin{pmatrix}
a_{ij}
\end{pmatrix}, \quad i, j \in (1, s).
\]
\[(23)\]
Clearly, making use of (13) and (23) leads to
\[
A = VA_sV^{-1}.
\]
\[(24)\]
Therefore, the weighting coefficients matrix \(A\) also satisfies \(V\)-transformation. It can be inferred from (22) that
\[
\bar{x}_i = x_n + h \sum_{j=1}^{s} a_{ij} f(t_n + c_jh, \bar{x}_j), \quad i \in (1, s).
\]
\[(25)\]
Since \( c_i = 1, t_n + c_i h = t_n + h = t_{n+1} \); therefore, \( \bar{x}_i \) \((i = s)\) is the approximate solution at the end of the step. Then, \( \bar{x}_i \) can be rewritten as the following form:

\[
\bar{x}_i = x_{n+1} = x_n + h \sum_{j=1}^{s} b_j f \left( t_n + c_j h, \bar{x}_j \right)
\]

(26)

where \( b_j = a_j, j \in (1, s) \). It can be seen that (25) and (26) are the standard forms for an \( s \)-stage Runge-Kutta method. Since \( \bar{x}_i = x_{n+1} \), the equivalent Runge-Kutta method is a reducible method [20]. In fact, the traditional differential quadrature method generally does not involve postprocessing, so the Runge-Kutta method converted from traditional differential quadrature method will naturally become a reducible method. The Runge-Kutta method can be conveniently summarized in the Butcher tableau [18] as

\[
\begin{bmatrix}
c_1 & A_1 \\
b_1 & b_T
\end{bmatrix}
\]

(27)

where \( b_T = (b_1, b_2, \ldots, b_s) = (a_{11}, a_{21}, \ldots, a_{ss}) \).

4. Analysis of the Basic Characteristics of the Differential Quadrature Method

The stability and accuracy characteristics of the equivalent Runge-Kutta method will be investigated next. From (13) and (23), it can be inferred that

\[
\begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1s} \\
a_{21} & a_{22} & \cdots & a_{2s} \\
\vdots & \vdots & \ddots & \vdots \\
a_{s1} & a_{s2} & \cdots & a_{ss}
\end{bmatrix}
\begin{bmatrix}
c_1 \\
c_2 \\
\vdots \\
c_s
\end{bmatrix}

= \begin{bmatrix}
1 & c_1 & c_1^2 & \cdots & c_1^{s-1} \\
1 & c_2 & c_2^2 & \cdots & c_2^{s-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & c_s & c_s^2 & \cdots & c_s^{s-1}
\end{bmatrix}
\begin{bmatrix}
1 \\
\frac{1}{2} \\
\vdots \\
\frac{1}{s}
\end{bmatrix}
\]

(28)

Equation (28) reduces to

\[
\sum_{j=1}^{s} a_j c_j^{k-1} = \frac{c_i^k}{k}, \quad i \in (1, s), \quad k \in (1, s).
\]

(29)

On the other hand, since \( b_j = a_j, j \in (1, s) \) and \( c_i = 1 \), from (28), it can be inferred that

\[
b_T V = \begin{bmatrix}
1 & \frac{1}{2} & \cdots & \frac{1}{s}
\end{bmatrix}
\]

(30)

Similarly, (30) reduces to

\[
\sum_{j=1}^{s} b_j c_j^{k-1} = \frac{1}{k}, \quad k \in (1, s).
\]

(31)

Obviously, it has been shown that the equivalent Runge-Kutta method at least satisfies simplifying assumptions \( C(s) \) and \( B(s) \) from (29) and (31). Furthermore, it can be verified that the equivalent Runge-Kutta method only satisfies simplifying assumptions \( D(s) \). From Theorem 5.1 on page 71 in [17], it can be concluded that the implicit Runge-Kutta method or the corresponding differential quadrature method is \( s \)-stage \( s \)-order.

The stability function \( R(z) \) of the equivalent Runge-Kutta method or the corresponding differential quadrature method is given by the formula [16–18]

\[
R(z) = \frac{\det(I + z (eb^T - A))}{\det(I --zA)},
\]

(32)

where, as usual, \( I \) is the identity matrix of dimension \( s \). Due to grid points’ asymmetric distribution, the equivalent implicit Runge-Kutta method is not a symmetric method. As a result, there is a unique adjoint method (also called reflected method) [16, 18], which is defined as \( c^* = e - P c \),

\[
P A^* P = \text{eb}^T - \text{A},
\]

(33)

where \( P \) is

\[
P = \begin{bmatrix}
1 \\
\vdots \\
1
\end{bmatrix} \in \mathbb{R}^{sx}.
\]

(34)

Furthermore, from Theorem 345B on page 221 in [18], if the original method satisfies the simplifying assumptions \( C(s) \) and \( B(s) \), then the adjoint method also satisfies the same simplifying assumptions. Hence, the adjoint method enjoys \( V \)-transformation:

\[
A^* = V^* A_s^* (V^*)^{-1}, \quad V^* = \begin{bmatrix}
1 & c_1^* & \cdots & (c_1^*)^{s-1} \\
1 & c_2^* & \cdots & (c_2^*)^{s-1} \\
\vdots & \vdots & \ddots & \vdots \\
1 & c_s^* & \cdots & (c_s^*)^{s-1}
\end{bmatrix},
\]

(35)

where \( A_s^* \) is

\[
A_s^* = \begin{bmatrix}
0 & 0 & 0 & \cdots & \beta_s \\
1 & 0 & 0 & \cdots & \beta_s \\
0 & \frac{1}{2} & \cdots & \cdots & \cdots \\
\vdots & \vdots & \ddots & \cdots & \cdots \\
0 & 0 & \cdots & \frac{1}{s-1} & \beta_s
\end{bmatrix}.
\]

(36)
with

\[ \beta_s = [\beta_1, \beta_2, \ldots, \beta_s]^T = \frac{1}{s} (V^*)^{-1} (c^*)^T, \]  

\[ (c^*)^T = [(c_1^*)^T, (c_2^*)^T, \ldots, (c_s^*)]^T. \]  

Equation (32) can be reduced to

\[ R(z) = \frac{\det(I + z(\mathbf{A}^* \mathbf{P}))}{\det(I - z(A_s^*)_s)} = \frac{\det(I + z\mathbf{A}_s^*)}{\det(I - z\mathbf{A}_s)} \]

\[ = 1 - \sum_{k=s}^1 \gamma_k ((k-1)!/(s-1)!)(-z)^{s-k+1} \]

\[ = 1 - \sum_{k=2}^{s+1} \gamma_k ((k-1)!/(s-1)!)(-z)^{s-k+1}. \]

(39)

Because \( \mathbf{A}_s^* \) and \( \mathbf{A}_s^* \) are a class of special matrices, (38) can be evaluated as

\[ R(z) = \frac{\det(I + z\mathbf{A}_s^*)}{\det(I - z\mathbf{A}_s)} \]

\[ = 1 - \sum_{k=s}^1 \beta_k ((k-1)!/(s-1)!)(-z)^{s-k+1}. \]

V-transformation of matrix \( \mathbf{A}^* \) is

\[ \mathbf{A}^* = (\mathbf{G}^*)^{-1} \mathbf{V} \]

\[ \mathbf{A}^* = \left( \begin{array}{ccc}
    1 & 0 & 0 \\
    1 & 1 & 1 \\
    1 & 2 & 3 \\
\end{array} \right) \]

\[ \left( \begin{array}{ccc}
    1 & 0 & 2 \\
    1 & 1 & 3 \\
    1 & 2 & 4 \\
\end{array} \right) \]

(45)

5. Improved Differential Quadrature Method

Based on the above deduction, the traditional differential quadrature method is a method of \( s \)-stage \( s \)-order. Compared with the multistage high-order Runge-Kutta method, for example, Gauss method (\( s \)-stage 2\( s \)-order method), it has the disadvantage of lower precision. As it is well-known that if the stability function of a numerical method is diagonal Padé approximations to the exponential function, then this method is the method of A-stable and \( s \)-stage 2\( s \)-order [22]. Inspired by this idea, the stability function of new Runge-Kutta method or new differential quadrature method have been converted into diagonal Padé approximation to the exponential function by using undetermined coefficients method. From (38), it can be seen that the stability function of the equivalent Runge-Kutta method will be determined by \( \mathbf{A}_s \) and \( \mathbf{A}_s^* \). Suppose \( \mathbf{A}_s = \mathbf{A}_s^* = \mathbf{A}_o \), without changing \( \mathbf{b} \) and \( \mathbf{V} \), a new Runge-Kutta method

\[ \mathbf{c} \]

\[ \mathbf{b}^T \]

Then, the stability function of this new Runge-Kutta method becomes

\[ R(z) = \frac{\det(I + z\mathbf{A}_o)}{\det(I - z\mathbf{A}_o)} = \frac{1 - \sum_{k=s}^1 \gamma_k ((k-1)!/(s-1)!)(-z)^{s-k+1}}{1 - \sum_{k=s}^1 \gamma_k ((k-1)!/(s-1)!)(z)^{s-k+1}}. \]

(47)
From (46) and (47), it can be inferred that the last column elements in \( \bar{A} \) determine the stability function. To improve the order of new Runge-Kutta method, undetermined coefficients \( \gamma \) can be selected so that the stability function of new Runge-Kutta method becomes the diagonal Padé approximations to the exponential function (defined by \( e^z \)):

\[
R(z) = \frac{1 - \sum_{k=1}^{s-1} y_k ((k-1)!/(s-1)!)(-z)^{s-k+1}}{1 - \sum_{k=3}^{s} y_k ((k-1)!/(s-1)!)(-z)^{s-k+1}} = e^{\gamma z}.
\]

(48)

By comparing the coefficients on both sides of (48), undetermined coefficients \( \gamma \) can be conveniently obtained as

\[
s = 2, \quad \gamma_2 = \left[ \begin{array}{c} -1/12 \\ 1/2 \end{array} \right],
\]

\[
s = 3, \quad \gamma_3 = \left[ \begin{array}{c} 1/60 \\ 1/5 \end{array} \right],
\]

\[
s = 4, \quad \gamma_4 = \left[ \begin{array}{c} -1/280 \\ 1/14 - 9/28 \\ 1/2 \end{array} \right],
\]

\[
\vdots
\]

After getting \( \gamma = (\gamma_1, \ldots, \gamma_s)^T \), coefficients matrix \( \bar{A} \) or \( \bar{G} \) can also be easily computed through using (46). Therefore, a class of new Runge-Kutta method of \( s \)-stage 2s-order has been successfully constructed. In other words, a class of improved differential quadrature method of \( s \)-stage 2s-order has been derived. Besides, the adjoint method of new Runge-Kutta method is also \( s \)-stage 2s-order.

Take the same as above, the improved differential quadrature method using Uniform grid points will be given as an example. It can be worked out that the new matrices \( \bar{G}_0 \) and \( \bar{G} \) are given by

\[
\bar{G}_0 = \left( \begin{array}{cccc} 4 & 4 \\ 3 & -2 \end{array} \right), \quad \bar{G} = \left( \begin{array}{ccc} -17/2 & 10 & -17/6 \\ 11/2 & 4 & 1/6 \\ 75/2 & -42 & 33/2 \end{array} \right).
\]

(50)

and the Butcher tableau of new Runge-Kutta method is

\[
c \left( \begin{array}{cccc} 1 & 73 & 23 & 13 \\ 3 & 120 & 60 & 120 \\ 2 & 97 & 17 & 17 \\ 3 & 120 & 60 & 120 \end{array} \right).
\]

(51)

V-transformation of matrix \( \bar{A} \) is

\[
\bar{A} = \left( \begin{array}{ccc} 1 & 1/3 & 1 \\ 1 & 2/3 & 4 \\ 1 & 1 \end{array} \right) \left( \begin{array}{ccc} 0 & 0 & 1/60 \\ 1 & 0 & -1/5 \\ 0 & 1/2 & 1 \end{array} \right) \left( \begin{array}{ccc} 1 & 1/3 & 1 \\ 1 & 2/3 & 4 \\ 1 & 1 \end{array} \right)^{-1}.
\]

(52)

6. Numerical Examples

Consider a two-degree-of-freedom system governed by

\[
\begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{pmatrix} + \begin{pmatrix} 6 & -2 \\ -2 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 10 \end{pmatrix},
\]

(55)

with initial condition

\[
x_1 = 0, \quad \dot{x}_1 = 0
\]

\[
x_2 = 0, \quad \dot{x}_2 = 0.
\]

(56)

The exact solution of this problem is

\[
x_1 = 1 - \frac{5}{3} \cos \left( \sqrt{2}t \right) + \frac{2}{3} \cos \left( \sqrt{5}t \right),
\]

\[
x_2 = 3 - \frac{5}{3} \cos \left( \sqrt{2}t \right) - \frac{4}{3} \cos \left( \sqrt{5}t \right).
\]

(57)

The differential quadrature method can be used to find the numerical solutions to transient response directly. The detailed calculation steps of solving second-order differential equations in [11] can be seen. Figures 1, 2, and 3 show the displacement error trajectories comparison of traditional differential quadrature method and improved differential quadrature method with the same step size \( h = 0.5s \). In these Figures, the exact solution at each step is used for comparison. From Figures 1–3, it is evident that improved differential quadrature method is two orders of magnitude higher than traditional differential quadrature method. The error of improved differential quadrature method ranges between \( 10^{-5} \) and \( 10^{-4} \), even with a large step size \( h = 0.5s \).

In order to compare the calculation precision of the traditional differential quadrature method and the improved differential quadrature method better, dynamic equations (55) have been computed for different time steps. Table 1 shows the computational results at \( t = 60 \) s. Comparing with
Figure 1: Error trajectories comparison of different DQM using Chebyshev grid ($s = 3, h = 0.5s$).

Figure 2: Error trajectories comparison of different DQM using Chebyshev-Gauss-Lobatto grid ($s = 3, h = 0.5s$).

Figure 3: Error trajectories comparison of different DQM using Uniform grid ($s = 3, h = 0.5s$).
Table 1: Computational results of the displacement (t = 60 s).

<table>
<thead>
<tr>
<th>s = 5</th>
<th>h = 1s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional DQM-Chebyshev</td>
<td>2.25973583</td>
</tr>
<tr>
<td>Improved DQM-Chebyshev</td>
<td>2.264406270</td>
</tr>
<tr>
<td>Traditional DQM-Chebyshev-Gauss-Lobatto</td>
<td>2.276013635</td>
</tr>
<tr>
<td>Improved DQM-Chebyshev-Gauss-Lobatto</td>
<td>2.26406270</td>
</tr>
<tr>
<td>Traditional DQM-Uniform</td>
<td>2.342513616</td>
</tr>
<tr>
<td>Improved DQM-Uniform</td>
<td>2.26406270</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>s = 10</th>
<th>h = 2s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional DQM-Chebyshev</td>
<td>2.264383332</td>
</tr>
<tr>
<td>Improved DQM-Chebyshev</td>
<td>2.264385552</td>
</tr>
<tr>
<td>Traditional DQM-Chebyshev-Gauss-Lobatto</td>
<td>2.264389479</td>
</tr>
<tr>
<td>Improved DQM-Chebyshev-Gauss-Lobatto</td>
<td>2.264386552</td>
</tr>
<tr>
<td>Traditional DQM-Uniform</td>
<td>2.265087814</td>
</tr>
<tr>
<td>Improved DQM-Uniform</td>
<td>2.264386561</td>
</tr>
</tbody>
</table>

the exact solution, the numerical results using traditional
differential quadrature method are the same until the first
decimal place when s = 5, h = 1s, while the numerical
results using improved differential quadrature method are the
same until the third decimal place. When s = 10, h = 2s,
the numerical results using traditional differential quadrature
method are the same until the fifth decimal place, while
the numerical results using improved differential quadrature
method are the same until the seventh decimal place (espe-
cially using Chebyshev grid points and Chebyshev-Gauss-
Lobatto grid points, the numerical results are almost the exact
solution).

7. Conclusion

In this paper, the linear stability and the order of differential
quadrature method in time domain are systematically studied
in detail and a class of new differential quadrature method
of s-stage 2s-order is proposed. From the above analysis and
derivation, the following conclusions can be made.

(1) Based on general polynomial as test functions,
the weighting coefficients matrix of the differential
quadrature method satisfies the V-transformation.
It plays an extremely important role in the analysis-
thesis of basic characteristics of differential quadrature
method and its improvement.

(2) The traditional differential quadrature method can be
converted into equivalent Runge-Kutta method of A-
stable and s-stage s-order. Therefore, compared with
the commonly used single-stage low-order numerical
integral methods, even with a small number of grid
points, the traditional differential quadrature method
also give more accurate solutions. This is the main
mechanism that differential quadrature method has
been successfully applied in many fields.

(3) Finally, by making the stability function of equiva-
 lent Runge-Kutta method become the diagonal Padé
approximations to the exponential function, a class
of improved differential quadrature method of s-
stage 2s-order and A-stable is proposed. Therefore,
the improved differential quadrature method can be
extended to multi-degree-of-freedom time domain
dynamic systems, which can produce higher accurate
solutions at lower computational cost.

Conflict of Interests

The authors declare that there is no conflict of interests
regarding the publication of this paper.

Acknowledgment

The authors gratefully acknowledge the support from the
National Natural Science Foundation of China (NSFC)
through its Grant 51377098.

References

integration,” Journal of Mathematical Analysis and Applications,

 technique for the rapid solution of nonlinear partial differential
equations,” Journal of Computational Physics, vol. 10, pp. 40–52,
1972.

quadrature and cubature methods vis-`a-vis some conventional
 techniques in context of convection-diffusion-reaction prob-
lems,” Chemical Engineering Science, vol. 50, no. 3, pp. 531–547,
1995.

computational mechanics: a review,” Applied Mechanics

quadrature to static analysis of structural components,” Interna-

for static and free vibration analyses of anisotropic plates,” Inter-

[7] F. Civan, ”Solving multivariable mathematical models by the
quadrature and cubature methods,” Numerical Methods for

differential quadrature method to structural problems,” Inter-
national Journal for Numerical Methods in Engineering, vol. 37,

quadrature to transport processes,” Journal of Mathematical


