Available online at www.sciencedirect.com
SCIENCE DIRECT
Journal of Sound and Vibration 296 (2006) 908–918
JOURNAL OF SOUND AND VIBRATION
www.elsevier.com/locate/jsvi

Solution of nonlinear initial-value problems by the spline-based differential quadrature method

Hongzhi Zhong*, Mengyu Lan

Department of Civil Engineering, Tsinghua University, Beijing 100084, China

Available online 2 June 2006

Abstract

The spline-based differential quadrature method (SDQM) is applied to the solution of nonlinear initial-value problems. Explicit expressions of weighting coefficients for approximation of derivatives are presented. Dynamic systems with Duffing-type nonlinearity are solved to demonstrate the effectiveness of the method. Numerical results of three examples show that the spline-based differential quadrature method is versatile and stable in the solution of nonlinear initial-value problems. It can be counted on to achieve satisfactory accuracy for long-term integration.

1. Introduction

The application of the differential quadrature method (DQM) to the solution of boundary-value problems has achieved great success. Comprehensive surveys of the DQM were given by Bert and Malik [1] and Shu [2]. Interestingly, the method was initially proposed for the solution of initial-value problems [3]. However, inadequate research has been conducted until recent years. One of the major developments of the application of DQM in the solution of initial-value problems is the ways in which the initial conditions are incorporated into the solution process. As pointed out by Fung [4], direct application of the conventional DQM results in unstable time integration scheme. An early attempt was made by Chen who introduced a transformation of time variable for second-order dynamic systems in which the initial conditions are incorporated simply and effectively [5]. Using Hermite interpolation to incorporate the initial conditions, Wu and Liu proposed a general differential quadrature rule [6,7] to solve initial-value problems. An implicit scheme where the time span is dicretized into a number of large blocks was proposed by Shu et al. [8]. This block marching technique reduces significantly the accumulation error since the DQM was applied to each block. In a series of papers on the solution of initial-value problems using the DQM, Fung introduced a modified DQM to incorporate initial conditions [4,9]. He also discussed at length the stability of various grid patterns in the DQM. It has been found [4] that among Legendre, Radau, Chebyshev, Chebyshev–Gauss–Lobatto and uniformly spaced grid patterns, only Legendre, Radau, Chebyshev grid patterns are unconditionally stable using the modified DQM. In addition, Tomasiello [10] developed an iterative DQM to handle initial-value problems. Despite these

*Corresponding author. Tel.: +86 10 62781891; fax: +86 10 62771132.
E-mail address: hzz@tsinghua.edu.cn (H. Zhong).

0022-460X/$ - see front matter
doi:10.1016/j.jsv.2006.03.018

developments, there has been limited information on the application of the DGM to long-term integration especially for nonlinear dynamic problems.

The spline-based differential quadrature is a ramification of the differential quadrature family. Taking the advantages of spline functions, the spline-based differential quadrature is free of the limitation of grid point number, which can be a bottleneck for the conventional DQM to achieve long-term integration. So far, differential quadrature based on quintic and sextic B-spline functions has been successfully applied to the solution of various boundary-value problems [11,12]. In this paper, a DQM based on cubic B-spline functions is introduced and used to solve second- and fourth-order initial-value differential equations with Duffing-type nonlinearity. The spline-based DQM exhibits more flexibility in selection of grid point number due to the local interpolation nature of splines, which is similar to that of the low-order finite difference method. The stability of the spline-based DQM is also discussed. Although the proposed spline-based DQM is conditionally stable, its well-behaved stability still makes it stand out. This is further exhibited in the excellent performance during the solution of nonlinear Duffing-type problems.

The present paper is organized into the following sections. In Section 2, the cardinal cubic spline interpolation functions are constructed and the explicit expressions of the weighting coefficients for derivative approximation are given. The incorporation of initial conditions, the stability of the spline-based DQM and two ways of application are addressed in Section 3. Three dynamic systems of Duffing-type nonlinearity are solved in Section 4. Discussion of the results is given in Section 5 and some concluding remarks are given in Section 6.

2. Differential quadrature based on cubic B-splines

Cubic B-splines are widely used in practice for approximation, because they provide a suitable balance between flexibility and accuracy [13,14]. In this paper, cubic B-splines are used to construct differential quadrature format for solution of initial-value problems.

2.1. Cardinal cubic spline interpolation

First of all, a set of uniformly spaced nodes is selected in a normalized interval [0,1], i.e.

$$x_0 = 0,\ x_N = 1,\ x_{j+1} - x_j = h, \quad j = 0, 1, 2, \ldots, N-1, N, \tag{1}$$

where $N+1$ is the number of nodes in the interval, h is the length of every segment. The normalized cubic B-spline function is given by [13,14]

$$\varphi_3(x) = \frac{1}{6h^3}\begin{cases} 0, & x \leqslant -2h; \\ (x+2h)^3, & -2h \leqslant x \leqslant -h; \\ (x+2h)^3 - 4(x+h)^3, & -h \leqslant x \leqslant 0; \\ (2h-x)^3 - 4(h-x)^3, & 0 \leqslant x \leqslant h; \\ (2h-x)^3, & h \leqslant x \leqslant 2h; \\ 0, & x \geqslant 2h. \end{cases} \tag{2}$$

Apparently, it is a piecewise polynomial, which covers four consecutive segments only. To construct a global interpolation function over the interval, usually extra nodes outside the interval $[x_0, x_N]$ are needed to meet the end condition requirements. A typical spline interpolation over the given interval using cubic B-splines is expressed in the form

$$s_3(x) = \sum_{j=-2}^{N+2} \Phi_j(x) y_j, \quad \Phi_j(x) = \Phi_0(x - jh). \tag{3}$$

In order to meet the required interpolation condition

$$s_3(x_i) = y_i, \tag{4}$$

the interpolation functions $\Phi_j(x)$ should satisfy the cardinal condition at every node, i.e.

$$\Phi_j(x_i) = \delta_{ij} = \begin{cases} 1, & i = j; \\ 0, & \text{otherwise}, \end{cases}$$
$$i,j = -2, -1, 0, 1, \ldots, N-1, N, N+1, N+2, \tag{5}$$

where $\Phi_j(x)$ are usually given in terms of a combination of translated and scaled spline function φ_3. To acquire the cardinal spline interpolation function $\Phi_j(x)$, the following three auxiliary spline interpolation functions [15] are constructed first:

$$\psi_3(x) = \sum_{j=-2}^{N+2} y_j \varphi_3(x - x_j), \tag{6a}$$

$$\psi_3^{\langle 1/2 \rangle}(x) = \sum_{j=-2}^{N+2} y_j \varphi_3^{\langle 1/2 \rangle}(x - x_j), \tag{6b}$$

$$\psi_3^{\langle 1 \rangle}(x) = \sum_{j=-2}^{N+2} y_j \varphi_3^{\langle 1 \rangle}(x - x_j), \tag{6c}$$

where

$$\varphi_3^{\langle 1/2 \rangle}(x) \equiv \varphi_3(x + h/2) + \varphi_3(x - h/2), \tag{7a}$$

$$\varphi_3^{\langle 1 \rangle}(x) \equiv \varphi_3(x + h) + \varphi_3(x - h). \tag{7b}$$

With the local non-zero property of the spline function $\varphi_3(x)$, all the terms but the one containing y_j on the right sides of Eq. (6) are eliminated. Thus, the cardinal spline interpolation function is obtained as

$$s_3(x) = \tfrac{10}{3}\psi_3(x) - \tfrac{4}{3}\psi_3^{\langle 1/2 \rangle}(x) + \tfrac{1}{6}\psi_3^{\langle 1 \rangle}(x). \tag{8}$$

Hence

$$\Phi_j(x) = \tfrac{10}{3}\varphi_3(x - x_j) - \tfrac{4}{3}\varphi_3^{\langle 1/2 \rangle}(x - x_j) + \tfrac{1}{6}\varphi_3^{\langle 1 \rangle}(x - x_j). \tag{9}$$

It has been shown that the accuracy of the cardinal spline in Eq. (8) is of order $O(h^4)$ [15]. It can be shown readily that Eq. (4) is satisfied at every node (see Fig. 1). Since the extra nodes outside the interval are often cumbersome to handle, non-integral nodes within the interval are introduced instead in this paper. Namely, $x_{1/2} = h/2$, $x_{3/2} = 3h/2$ and $x_{N-3/2} = (N - 3/2)h$, $x_{N-1/2} = (N - 1/2)h$ are added in the vicinity of the two ends of the interval. As a result, the spline-based differential quadrature meets the self-starting requirement of a good algorithm for initial-value problems [16]. The function values at the non-integral nodes are given by

$$y_{1/2} = \sum_{i=-2}^{N+2} \Phi_i(h/2) y_i = \frac{1}{288}(y_{-2} + y_3) - \frac{7}{96}(y_{-1} + y_2) + \frac{41}{72}(y_0 + y_1) \tag{10a}$$

and

$$y_{3/2} = \sum_{i=-2}^{N+2} \Phi_i(3h/2) y_i = \frac{1}{288}(y_{-1} + y_4) - \frac{7}{96}(y_0 + y_3) + \frac{41}{72}(y_1 + y_2). \tag{10b}$$

Then, the function values at the extra nodes are solved for from Eqs. (10a) and (10b)

$$y_{-1} = 21(y_0 + y_3) - y_4 + 288 y_{3/2} - 164(y_1 + y_2), \tag{11a}$$

$$y_{-2} = 277 y_0 + 288 y_{1/2} - 3608 y_1 + 6048 y_{3/2} - 3423 y_2 + 440 y_3 - 21 y_4. \tag{11b}$$

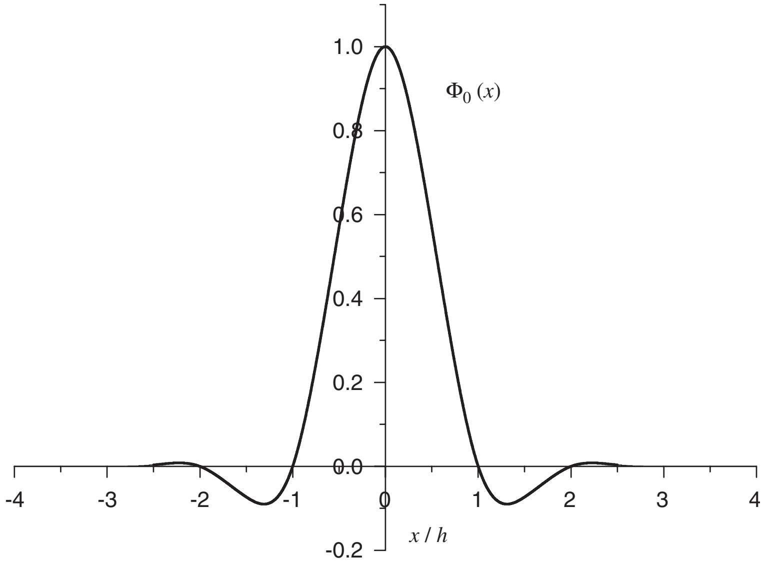

Fig. 1. A typical cubic cardinal spline function.

Similarly, the function values at the extra nodes outside the right end of the interval are expressed as

$$y_{N+1} = 21(y_{N-3} + y_N) - y_{N-4} + 288y_{N-3/2} - 164(y_{N-2} + y_{N-1}), \tag{12a}$$

$$y_{N+2} = 277y_N + 288y_{N-1/2} - 3608y_{N-1} + 6048y_{N-3/2} - 3423y_{N-2} + 440y_{N-3} - 21y_{N-4}. \tag{12b}$$

Now, the cardinal cubic B-spline interpolation function is re-arranged into the following form which is free of extra outside nodes

$$s_3(x) = \sum_{j=0}^{N} \Omega_j(x)y_j, \quad \Omega_j(x_i) = \delta_{ij} = \begin{cases} 1, & i = j; \\ 0, & \text{otherwise}, \end{cases}$$
$$i,j = 0, 1/2, 1, 3/2, 2, \ldots, N-2, N-3/2, N-1, N-1/2, N, \tag{13}$$

where

$$\begin{aligned}
\Omega_0(x) &= 277\Phi_{-2}(x) + 21\Phi_{-1}(x) + \Phi_0(x), \\
\Omega_{1/2}(x) &= 288\Phi_{-2}(x), \\
\Omega_1(x) &= -3608\Phi_{-2}(x) - 164\Phi_{-1}(x) + \Phi_1(x), \\
\Omega_{3/2}(x) &= 6048\Phi_{-2}(x) + 288\Phi_{-1}(x), \\
\Omega_2(x) &= -3423\Phi_{-2}(x) - 164\Phi_{-1}(x) + \Phi_2(x), \\
\Omega_3(x) &= 440\Phi_{-2}(x) + 21\Phi_{-1}(x) + \Phi_3(x), \\
\Omega_4(x) &= -21\Phi_{-2}(x) - \Phi_{-1}(x) + \Phi_4(x),
\end{aligned} \tag{14a}$$

$$\Omega_i(x) = \Phi_i(x) \quad \text{for} \quad 5 \leqslant i \leqslant N-5 \tag{14b}$$

and

$$\begin{aligned}
\Omega_{N-4}(x) &= -21\Phi_{N+2}(x) - \Phi_{N+1}(x) + \Phi_{N-4}(x),\\
\Omega_{N-3}(x) &= 440\Phi_{N+2}(x) + 21\Phi_{N+1}(x) + \Phi_{N-3}(x),\\
\Omega_{N-2}(x) &= -3423\Phi_{N+2}(x) - 164\Phi_{N+1}(x) + \Phi_{N-2}(x),\\
\Omega_{N-3/2}(x) &= 6048\Phi_{N+2}(x) + 288\Phi_{N+1}(x),\\
\Omega_{N-1}(x) &= -3608\Phi_{N+2}(x) - 164\Phi_{N+1}(x) + \Phi_{N-1}(x),\\
\Omega_{N-1/2}(x) &= 288\Phi_{N+2}(x),\\
\Omega_{N}(x) &= 277\Phi_{N+2}(x) + 21\Phi_{N+1}(x) + \Phi_{N}(x).
\end{aligned} \tag{14c}$$

It is noted that the selection of non-integral nodes at the vicinity of the two ends is not unique. The selection of different non-integral nodes at the vicinity of the two ends should meet the criterion that the extra nodes outside the domain can be expressed in terms of the introduced nodes as well as the inner nodes at the vicinity of the two ends.

2.2. Weighting coefficients for spline-based differential quadrature

The essence of the differential quadrature is that the derivative of a function with respect to a space variable at a given point is approximated by a weighted linear summation of the function values at all discrete nodes in the domain. Therefore, the approximation of a derivative at a node in spline-based differential quadrature is given by

$$\begin{aligned}
&D_n\{f(x)\}_i = \sum_{j=0}^{N} C_{ij}^{(n)} f(x_j),\\
&i,j = 0, 1/2, 1, 3/2, 2, \ldots, N-2, N-3/2, N-1, N-1/2, N,
\end{aligned} \tag{15}$$

where D_n is a differential operator of order n, the subscript i indicates the value of $D_n\{f(x)\}$ at node x_i, $C_{ij}^{(n)}$ are the weighting coefficients related to the function values $f(x_j)$. In the spline-based DQM, it is required that Eq. (15) be exactly satisfied when function $f(x)$ takes the cardinal spline basis functions $\Omega_j(x)$. Consequently, all weighting coefficients are given in explicit forms

$$\begin{aligned}
&C_{ij}^{(1)} = \Omega_j^{(1)}(x_i), \quad C_{ij}^{(2)} = \Omega_j^{(2)}(x_i),\\
&i,j = 0, 1/2, 1, 3/2, 2, \ldots, N-2, N-3/2, N-1, N-1/2, N.
\end{aligned} \tag{16}$$

It is worth mentioning that the localized nature of splines results in banded weighting coefficient matrices for derivatives. Meanwhile, the following relationships among weighting coefficients exist

$$\begin{aligned}
&C_{ij}^{(1)} = -C_{(N-i)(N-j)}^{(1)}, \quad C_{ij}^{(2)} = C_{(N-i)(N-j)}^{(2)},\\
&i,j = 0, 1/2, 1, 3/2, 2, \ldots, N-2, N-3/2, N-1, N-1/2, N.
\end{aligned} \tag{17}$$

3. Application of SDQM to initial-value problems

3.1. Incorporation of initial conditions

With the first-order weighting coefficients $C_{ij}^{(1)}$, initial conditions are incorporated easily into the spline-based differential quadrature adopting the same strategy as [9]. For simple representation, the superscript of the weighting coefficients for the first-order derivatives is omitted. Then, Eq. (15) is written as

$$s'(x_i) = D_1\{f(x)\}_i = \sum_{j=0}^{N} C_{ij} f(x_j) = C_{i0} f(x_0) + \sum_{j=1/2}^{N} C_{ij} f(x_j). \tag{18}$$

Likewise, the second-order derivative of the function is written as

$$\begin{aligned} s''(x_i) &= C_{i0}f'(x_0) + \sum_{j=1/2}^{N} C_{ij}(C_{j0}f(x_0) + \sum_{k=1/2}^{N} C_{jk}f(x_k)) \\ &= C_{i0}f'(x_0) + \sum_{j=1/2}^{N} C_{ij}C_{j0}f(x_0) + \sum_{j=1/2}^{N} C_{ij} \sum_{k=1/2}^{N} C_{jk}f(x_k). \end{aligned} \tag{19}$$

Obviously, the two initial conditions $f(x_0)$ and $f'(x_0)$ are imposed.

Eqs. (18) and (19) can be rewritten in the matrix forms as follows:

$$\begin{Bmatrix} \dot{y}_{1/2} \\ \vdots \\ \dot{y}_N \end{Bmatrix} = \{C_0\}y_0 + [C]\begin{Bmatrix} y_{1/2} \\ \vdots \\ y_N \end{Bmatrix}, \tag{20}$$

$$\begin{Bmatrix} \ddot{y}_{1/2} \\ \vdots \\ \ddot{y}_N \end{Bmatrix} = \{C_0\}\dot{y}_0 + [C]\{C_0\}y_0 + [C][C]\begin{Bmatrix} y_{1/2} \\ \vdots \\ y_N \end{Bmatrix}, \tag{21}$$

where $\{C_0\}$ denotes $\{C_{i0}\}$ vector, $[C]$ denotes matrix $[C_{ij}]$ which is of size $(N+4)\times(N+4)$. In the same manner, high-order derivatives are given as

$$\begin{Bmatrix} y_{1/2}^{(m)} \\ \vdots \\ y_N^{(m)} \end{Bmatrix} = \sum_{p=0}^{m-1}[C]^{m-p-1}\{C_0\}y_0^{(p)} + [C]^m\begin{Bmatrix} y_{1/2} \\ \vdots \\ y_N \end{Bmatrix}, \quad m\geqslant 1. \tag{22}$$

The above way to incorporate the initial conditions is simple and straightforward. In addition, the recursive formula in Eq. (22) implies that low-order spline interpolation functions are applicable to the solution of high-order differential equations. This will be demonstrated later in an example.

3.2. Two ways of application

Since the spline-based differential quadrature is constructed on the normalized interval [0,1] and the method is virtually devoid of limitation on the grid point numbers. There are two practical means of applying the method to the solution of initial-value problems.

3.2.1. Indirect approach

This approach is conventional since most time integration methods adopt the similar procedures. It is characterized by the division of the entire time domain into several subintervals. The spline-based differential quadrature is applied to each subinterval. The end state conditions for each subinterval are used as the initial conditions for the next subinterval. The process is repeated until the solution at the end of the interval is obtained.

3.2.2. Direct approach

It has been rare to evaluate the state variables at a time directly unless it is close to the initial time. The development of spline-based differential quadrature, however, has made this achievable. This is attributed to the local behavior of the spline functions. Suppose that τ represents the normalized time domain [0,1], which is required by the spline-based differential quadrature; t represents the actual time variable, which is given as

$$t = L\tau, \quad \tau \in [0,1], \tag{23}$$

where L represents the length of actual time domain. Before the implementation of differential quadrature analog, all time derivatives are transformed onto the normalized time domain

$$\frac{d^{(n)}y}{dt^{(n)}} = \frac{1}{L^{(n)}}\frac{d^{(n)}y}{d\tau^{(n)}}, \quad n = 1, 2 \ldots . \tag{24}$$

3.3. Stability of spline-based differential quadrature

The stability of an algorithm is of critical concern in the solution of initial-value problems. Thus, an undamped linear single degree-of-freedom system is considered to investigate the stability of the present spline-based DQM. The dynamic equation is given as

$$\ddot{y}(t) + \omega^2 y(t) = 0 \tag{25}$$

with initial conditions

$$y(t = 0) = y_0 \text{ and } \dot{y}(t = 0) = v_0. \tag{26}$$

The stability is evaluated by calculating the spectral radius of the numerical amplification matrix [Z], which relates the state at the end of the temporal interval to the initial state, i.e.

$$\begin{Bmatrix} y|(t = \text{end}) \\ \dot{y}|(t = \text{end}) \end{Bmatrix} = [Z]\begin{Bmatrix} y_0 \\ v_0 \end{Bmatrix}. \tag{27}$$

The algorithm is said to be stable if the spectral radius $|\lambda|_{\max} \leqslant 1$ or unstable if otherwise.

Fig. 2 shows the spectral radii for spline-based differential quadrature with $N = 8$, 10 and 20. Clearly, there is no bifurcation or bubble in the curve because the eigenvalues of the amplification matrix are a complex conjugate pair. It is seen that the spline-based differential quadrature is conditionally stable. The unstable computational time interval falls within $[0.2, 1.3]\omega t$. With the increase of number of nodes, however, the unstable interval of the spectral radii remains virtually unaltered and even becomes narrower. As reported by Fung in his comprehensive research about the stability of the conventional DQM [4], the stable time domain varies with the increase of grid point number. Even for the differential quadrature with Legendre, Radau,

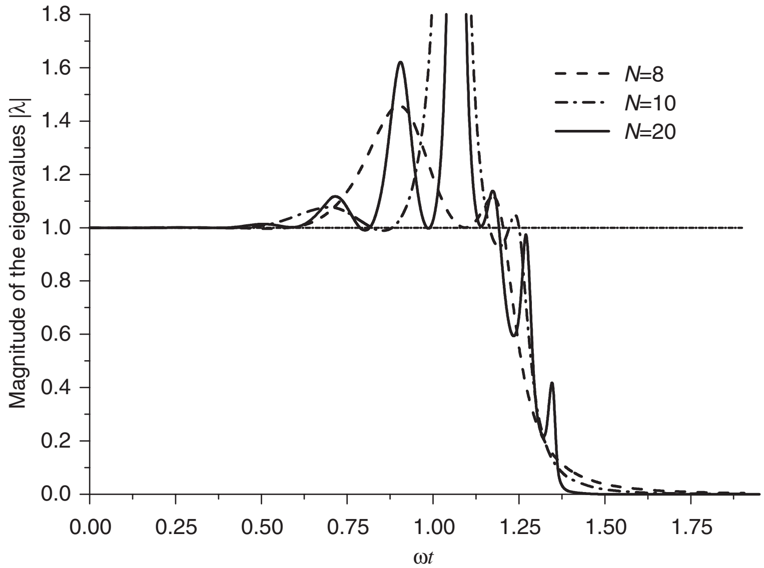

Fig. 2. Spectral radius of amplification matrix for $N = 8$, 10 and 20.

Chebyshev grid patterns, A-stability is not guaranteed if the grid number is large. In contrast, the stability criterion for spline-based differential quadrature is virtually invariant against the number of grid points. Furthermore, there does not exist limitation on the number of grid points in spline-based differential quadrature. This salient advantage makes it possible to achieve long-term integration with satisfactory accuracy regardless of the stability of the method. An example will be given later in which up to 800 nodes are used to acquire solution of the long-term solution of a dynamic system.

4. Duffing-type nonlinear equation

The form of Duffing-type nonlinear differential equation is written as

$$y^{(2)} + y + \varepsilon y^3 = F \sin \omega t, \tag{28}$$

where F and ε are given parameters, ω is also a given constant which represents the enforcing frequency. The analytical solution is given approximately by means of a trigonometric series [17]

$$y = a_1 \sin \omega t + a_3 \sin 3\omega t + a_5 \sin 5\omega t + \dots. \tag{29}$$

Three examples of Duffing-type nonlinear equation are calculated to illustrate the spline-based DQM.

Example 1. A second-order differential equation which represents the free vibration of pendulum is studied. The frequency of the oscillations depends on the initial displacement of the pendulum. Given parameters $\varepsilon = -1/6$, $\omega = 0.7$ and $F = 0$ in Eq. (28) for the unforced Duffing system, the solution [17] is

$$y \cong 2.058 \sin 0.7t + 0.0816 \sin 2.1t + 0.00337 \sin 3.5t \tag{30}$$

for initial conditions

$$y = 0, \quad y^{(1)} = 1.62376. \tag{31}$$

Example 2. Given parameters $F = 2$, $\varepsilon = -1/6$, and $\omega = 1$ for the forced Duffing equation, the solution [17] is

$$y \cong -2.5425 \sin t - 0.07139 \sin 3t - 0.00219 \sin 5t \tag{32}$$

for initial conditions

$$y = 0, \quad y^{(1)} = -2.7676. \tag{33}$$

Example 3. The following high-order (fourth-order) nonlinear differential equation is studied

$$y^{(4)} + 5y^{(2)} + 4y - \tfrac{1}{6}y^3 = 0. \tag{34}$$

Its analytical solution [17] for $\omega = 0.9$ is given as

$$y \cong 2.1906 \sin 0.9t - 0.02247 \sin 2.7t + 0.000045 \sin 4.5t \tag{35}$$

for initial conditions

$$y = 0, \quad y^{(1)} = 1.91103, \quad y^{(2)} = 0, \quad y^{(3)} = -1.15874. \tag{36}$$

The computational time domain is chosen as [0,100] for the three examples unless otherwise stated. The selection of long time domain is to investigate the stability and reliability of the method. As mentioned above, the accuracy of the conventional DQM decreases with the increase of domain length. For the above three examples, acceptable solutions over only a few cycles were obtained using the conventional DQM. In the forced Duffing equation case, convergent results were only obtainable for time domain shorter than one half cycle [7].

5. Numerical results and discussion

During the solution of dynamic systems of Duffing-type nonlinearity, the resultant nonlinear algebraic equations are solved using modified Powell hybrid algorithm [18]. To demonstrate the stability of the spline-based DQM, the indirect approach is adopted in the solution of unforced Duffing system with $\varepsilon = -1/6$, $\omega = 0.7$ and $F = 0$. The solution over 100 s (approximately 11 cycles) is sought and the time domain is divided into 100 subintervals. $N = 8$ is chosen for each subinterval. The results of displacement, velocity and acceleration, which are displayed in Fig. 3, are in excellent agreement with the analytical solution. With the indirect approach, solution with the same order of accuracy as shown in Fig. 3 over much longer time domain, say over 30 cycles, is also obtainable. When more subintervals are used in the indirect approach, larger accumulated errors result eventually. There is a tradeoff between the number of subintervals over the time domain and the number of points N. When fewer intervals are used in the indirect approach and the accumulation error is insignificant, the accuracy of the results is virtually the same as that of the direct approach. In the extreme case, i.e. the direct approach, large N is usually needed to gain satisfactory accuracy. The results with the same accuracy as shown in Fig. 3 are obtained using the direct approach when N is increased to 400. When more intervals are used in the indirect approach and the accumulation error becomes pronounced, the result accuracy differs for the two approaches. For clarity of presentation, only the results over the last few cycles are given in Figs. 3–5.

The forced dynamic system in Eq. (28) with $F = 2$, $\varepsilon = -1/6$, and $\omega = 1$ is dealt with using the indirect approach first. As reported in [7], this problem poses challenge to the conventional DQM since only one-half-cycle solutions of satisfactory accuracy were obtainable. It is found that results of satisfactory accuracy are obtained only for rather short time domain, say, $t \in [0, 10]$. To acquire the solution with good accuracy over

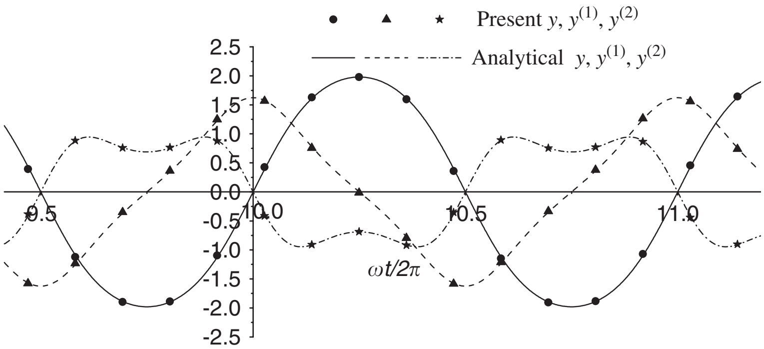

Fig. 3. Solution of Duffing equation of Example 1.

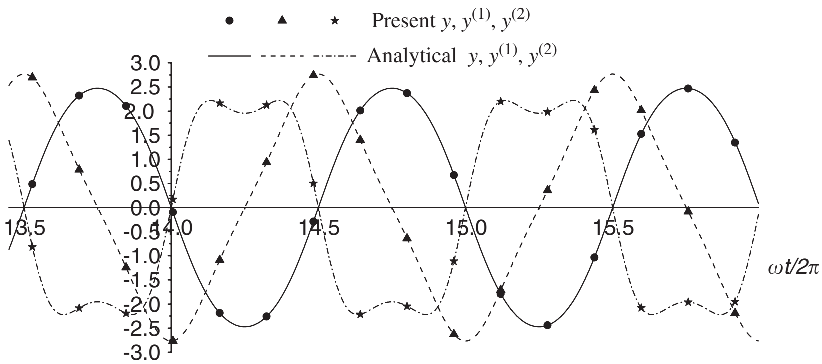

Fig. 4. Solution of Duffing equation of Example 2.

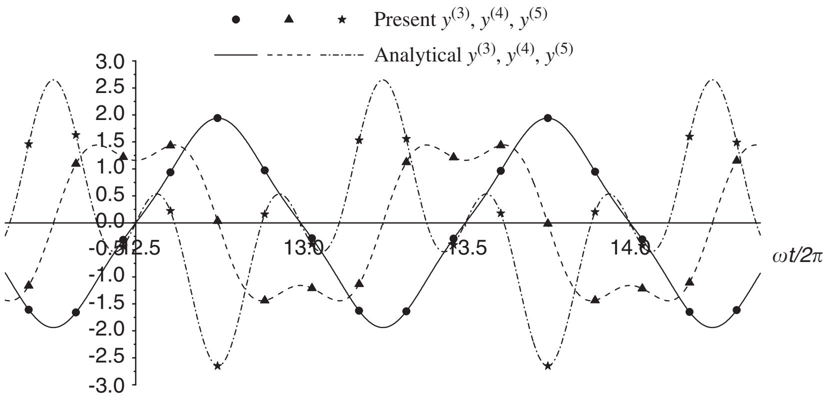

Fig. 5. Solution of Duffing equation of Example 3.

long time domain $t \in [0, 100]$, the direct approach and sufficient grid points are needed. When N is increased to 600, the results are in very good agreement with the analytical solution. The solutions of $y, y^{(1)}, y^{(2)}$ over $t \in [0, 100]$ (about 16 cycles) for $N = 800$ are shown in Fig. 4. It is noteworthy that difficulty in choosing a proper initial solution vector may arise when very large N is used in the direct approach. But this is compensated by the removal of possible accumulation error and stability concern.

To further demonstrate the spline-based DQM, the high-order system given in Eq. (34) is tackled. The solution over 100 s (approximately 14 cycles) is sought and the time domain is divided into 10 subintervals. The indirect approach is adopted and $N = 70$ is chosen in each subinterval. Very good agreement with the available analytical solution is reached, as shown in Fig. 5. This confirms the assertion that low-order spline functions can be used to solve high-order differential equations. Actually, solution with reasonable accuracy is obtainable when N takes 40. In addition, further reduction of the number of subintervals results in the need of large N and accordingly the requirement of good initial solution in the iterative process.

6. Concluding remarks

Based on the construction of cubic cardinal spline functions, a differential quadrature method (DQM) has been developed and applied to the solution of dynamic systems governed by Duffing-type nonlinear differential equations. Two solution approaches have been presented and their effectiveness has been verified. The stability of the spline-based DQM has also been studied. Although it is conditionally stable, the virtually invariant stable criterion with respect to the number of grid points makes it stand out as a promising numerical tool in the solution of initial-value problems. In addition, low-order spline functions are applicable to the solution of high-order differential equations. In particular, the proposed direct approach is attractive when long-term integration is encountered in the analysis of dynamic systems.

References

[1] C.W. Bert, M. Malik, Differential quadrature method in computational mechanics: a review, *Applied Mechanics Reviews* 49 (1996) 1–27.

[2] C. Shu, *Differential Quadrature and its Application in Engineering*, Springer, London, 2000.

[3] R.E. Bellman, J. Casti, Differential quadrature and long-term integration, *Journal of Mathematical Analysis and Applications* 34 (1971) 235–238.

[4] T.C. Fung, Stability and accuracy of differential quadrature method in solving dynamics problems, *Computer Methods in Applied Mechanics and Engineering* 191 (2002) 1311–1331.

[5] W. Chen, M. Tanaka, A study on time schemes for DRBEM analysis of elastic impact wave, *Computational Mechanics* 28 (2002) 331–338.

[6] T.Y. Wu, G.R. Liu, The generalized differential quadrature rule for initial-value differential equations, *Journal of Sound and Vibration* 233 (2000) 195–213.

[7] G.R. Liu, T.Y. Wu, Numerical solution for differential equations of Duffing-type non-linearity using the generalized differential quadrature, *Journal of Sound and Vibration* 237 (2000) 805–817.
[8] C. Shu, Q. Yao, K.S. Yeo, Block-marching in time with DQ discretization: an efficient method for time-dependent problems, *Computer Methods in Applied Mechanics and Engineering* 191 (2002) 4587–4597.
[9] T.C. Fung, Solving initial value problems by differential quadrature method–Part Two: second- and higher-order equations, *International Journal for Numerical Methods in Engineering* 50 (2001) 1429–1454.
[10] S. Tomasiello, Simulating non-linear coupled oscillators by an iterative differential quadrature method, *Journal of Sound and Vibration* 265 (2003) 507–525.
[11] H. Zhong, Spline-based differential quadrature for fourth order differential equations and its application to Kirchhoff plates, *Applied Mathematical Modelling* 28 (2004) 353–366.
[12] H. Zhong, Q. Guo, Vibration analysis of rectangular plates with free corners using spline-based differential quadrature, *Shock and Vibration* 11 (2004) 119–128.
[13] M.H. Schultz, *Spline Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1973.
[14] L.L. Schumaker, *Spline Functions: Basic Theory*, Wiley, New York, 1981.
[15] Y.S. Li, D.X. Qi, *Spline Function Method*, Science Press, Beijing, 1979 (in Chinese).
[16] T.J.R. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1987.
[17] F.R. Groves Jr., Numerical solution of nonlinear differential equations using computer algebra, *International Journal of Computer Mathematics* 13 (1983) 301–309.
[18] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, *Numerical Recipes—The Art of Scientific Computing*, Cambridge University Press, Cambridge, 1986.