Available online at www.sciencedirect.com

ScienceDirect

Journal of Sound and Vibration 314 (2008) 246–253

JOURNAL OF SOUND AND VIBRATION

www.elsevier.com/locate/jsvi

An assessment of the differential quadrature time integration scheme for nonlinear dynamic equations

Jian Liu, Xinwei Wang*

College of Aerospace Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, People's Republic of China

Received 15 September 2007; received in revised form 8 November 2007; accepted 4 January 2008
Handling Editor: L.G. Tham
Available online 11 February 2008

Abstract

In 1996, Xie [An assessment of time integration schemes for non-linear dynamic equations, *Journal of Sound and Vibration* 192(1) (1996) 321–331] presented an assessment on seven existing and commonly used time integration schemes for nonlinear dynamic equations. In this work, the differential quadrature (DQ) time integration scheme proposed by Fung in 2001 is assessed following the same procedures as Xie's. It is shown that accurate numerical results can be obtained by the DQ method when using much larger time step over the commonly used time integration schemes. Based on the results reported herein, some conclusions are drawn.

1. Introduction

In 2001, Fung [1,2] presented a new time integration scheme based on the differential quadrature (DQ) method. Based on the numerical examples, Fung showed that the proposed time integration scheme is unconditionally stable, higher-order accurate and computationally efficient for initial value problems. Thus, the DQ time integration scheme could be a powerful candidate for use in engineering practice. Earlier, Xie [3] performed an assessment on the seven commonly used time integration schemes, including the method of central finite difference, the Wilson-θ method, the Newmark method, the Houbolt method, the α-method of Hilber–Hughes–Taylor, together with the fourth-order Runge–Kutta method for the purpose of comparisons. Four simple nonlinear dynamic systems are tested. It is shown that some of the time integration schemes are not suitable for time integration over long time duration, since the numerical results are so inaccurate.

In this paper, the DQ time integration scheme, not included in Xie's paper, is tested following the same procedures as Xie's. Comparisons are made with existing results and some conclusions are drawn based on the results reported herein.

*Corresponding author.
E-mail address: wangx@nuaa.edu.cn (X. Wang).

0022-460X/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jsv.2008.01.004

2. A brief description of DQ time integration scheme

For completeness, the DQ time integration scheme in Refs. [1,2] is briefly described. According to the DQ method in time domain, $u(t)$ is expressed as

$$u(t)=\sum_{j=0}^{N} l_j(t)u_j, \quad t\in[0,\Delta t], \tag{1}$$

where $u_j=u(t_j)$, $(N+1)$ is the total number of grid points, $t_j(j=0,1,2,\ldots,N)$ are the grid point in [0,Δt] and $l_j(t)$ is the well-known Lagrangian interpolation functions, respectively.

Using Eq. (1) and taking the first derivative with respect to time t yield

$$\frac{\mathrm{d}u(t)}{\mathrm{d}t}=\dot{u}(t)=\sum_{j=0}^{N}\dot{l}_j(t)u_j, \quad t\in[0,\Delta t], \tag{2}$$

where the over dot denotes the first derivative with respect to time t. From Eq. (2) one obtains

$$\dot{u}_i=\dot{u}(t_i)=\sum_{j=0}^{N}\dot{l}_j(t_i)u_j=\sum_{j=0}^{N}G_{ij}u_j \quad (i=0,1,...,N), \tag{3}$$

where G_{ij} is the weighting coefficient of the first derivative with respect to time t and can be computed explicitly by [4,5]

$$G_{ij}=\frac{\omega'_N(t_i)}{(t_i-t_j)\omega'_N(t_j)} \quad (i\neq j), \quad G_{ii}=\sum_{j=0,i\neq j}^{N}\frac{1}{(t_i-t_j)}. \tag{4}$$

In Eq. (4), $\omega'_N(t_i)$ and $\omega'_N(t_j)$ are computed by

$$\omega'_N(t_i)=(t_i-t_0)(t_i-t_1)\cdots(t_i-t_{i-1})(t_i-t_{i+1})\cdots(t_i-t_{N-1})(t_i-t_N) \tag{5}$$

$$\omega'_N(t_j)=(t_j-t_0)(t_j-t_1)\cdots(t_j-t_{j-1})(t_j-t_{j+1})\cdots(t_j-t_{N-1})(t_j-t_N). \tag{6}$$

Let $\{u\}^{\mathrm{T}}=\lfloor u_1 \ \ldots \ u_N \rfloor$, $\{\dot{u}\}^{\mathrm{T}}=\lfloor \dot{u}_1 \ \ldots \ \dot{u}_N \rfloor$, all equations except for the first one in Eq (3), namely, i taking the value of $1,2,\ldots,N$, can be re-written by the following matrix form:

$$\{\dot{u}\}=\{G_0\}u_0+[G]\{u\}. \tag{7}$$

The second-order derivative with respect to time t can be expressed in terms of u_0, v_0, and $\{u\}$ in a similar way. Using Eq. (7), one has

$$\begin{aligned}\{\ddot{u}\}&=\{G_0\}\dot{u}_0+[G]\{\dot{u}\}=\{G_0\}v_0+[G]\{G_0\}u_0+[G][G]\{u\}\\&=\{G_0\}v_0+\{GG_0\}u_0+[GG]\{u\},\end{aligned} \tag{8}$$

where $\{\ddot{u}\}^{\mathrm{T}}=\lfloor \ddot{u}_1 \ \ldots \ \ddot{u}_N \rfloor$.

For unconditionally stable, higher-order accurate and computationally efficient, Fung [2] used the abscissa of Gaussian quadrature together with $t_0=0$ as the grid points, namely, t_j is computed by

$$t_j=\Delta t(1+x_j)/2 \qquad (j=1,2,\ldots,N), \tag{9}$$

where x_j are the abscissa of Gaussian quadrature in [−1,1]. For example, x_j takes the values of $-\sqrt{0.6},0,\sqrt{0.6}$ if $N=3$. It is seen that only one end point ($t=0$) of a time interval is assigned as a grid point, and the other one ($t=\Delta t$) of a time interval is not used as a grid point in determining the weighting coefficients, different from the conventional DQ method.

For a given second-order initial value problem, u_0 and $v_0=\dot{u}_0$ are known. Solving the resultant algebraic equations yields the vector $\{u\}$. At $t=\Delta t$, $u_{\Delta t}$ and $v_{\Delta t}=\dot{u}_{\Delta t}$ can be computed by [2]

$$u_{\Delta t}=u(t=\Delta t)=\sum_{j=0}^{N}l_j(t=\Delta t)u_j, \quad v_{\Delta t}=\frac{\mathrm{d}u(t=\Delta t)}{\mathrm{d}t}=\sum_{j=0}^{N}l_j(t=\Delta t)\dot{u}_j, \tag{10}$$

where $\{\dot{u}\}$ is obtained by Eq. (7).

In the DQ time integration scheme, $u_{\Delta t}$ and $v_{\Delta t}$ are the given initial conditions (u_0 and v_0) for the next time step. More details may be found in Refs. [1,2].

3. Numerical examples and discussion

The four examples considered in Ref. [3] are analyzed by using the DQ time integration scheme. Results are compared with other time integration schemes presented in Ref. [3]. To save the space, only a part of results in Ref. [3] are included for comparisons.

Example 1. Consider a hardening elastic spring. The nonlinear dynamic equation is

$$\ddot{u} + S_1 u(1 + S_2 u^2) = 0, \tag{11}$$

where $S_1 > 0, S_2 > 0$. The exact total energy is

$$E = \tfrac{1}{4}(2\dot{u}^2 + 2S_1 u^2 + S_1 S_2 u^4). \tag{12}$$

To assess the time integration scheme, the percent error in terms of the energy is introduced [3],

$$\mathrm{Er} = |(E - E_0)/E_0| \times 100\%, \tag{13}$$

where E_0 is the total energy at $t = 0$.

For numerical results present herein, $S_1 = 100$, $S_2 = 10$. The initial conditions are $u_0 = 1.5$ and $v_0 = 0.0$.

Table 1 lists the maximum percentage errors of the total energy over the time duration of $100T$. Except for the DQ results, all other data are directly cited from Ref. [3] for comparisons. In the table, the average acceleration method is equivalent to the Newmark method with $\beta = 0.25$, $\gamma = 0.5$, the method of central difference is equivalent to the Newmark method with $\beta = 0.0$, $\gamma = 0.5$, and Newmark(*) represents the Newmark method with $\beta = 0.3025$, $\gamma = 0.0$. The DQ results are obtained by the DQ method with $N = 3$ and direct iteration, instead of the Newton–Raphson method, is used in solving the nonlinear algebraic equations. During the iteration, the iteration stops if error $\leqslant \mathrm{eps} = 10^{-10}$, and then go to the next time step. The error is defined by error $= \sqrt{\sum_{k=1}^{N}(u_k^{i+1} - u_k^i)^2}$ for the DQ method, here i is the iteration number. It is seen from Table 1 that the DQ method is the best integration scheme among all methods listed in the table. Accurate results can be obtained with much larger time increment.

The phase portraits of the exact solution and the DQ solution with $\Delta t = T/20$ are given in Fig. 1a and b. It is seen that good agreement is obtained. According to Xie [3], only the average acceleration method and the method of central difference can give reasonable good results with the same time increment. Some of the results by the remaining investigated methods have deviated far from the exact solutions. This can also be seen from the maximum percentage errors of the total energy listed in Table 1. For $\Delta t = T/1000$ and the time duration of $100T$, the CPU time on the authors' Personal Computer by the DQ method and the average acceleration method together with the direct iteration method is approximately 0.063 and 0.032 s, respectively. It is noted that the re-calculated CPU time of the average acceleration method (0.032 s) is much less than the one (1.48 s) given by Xie [3]. The main reason to cause the difference is perhaps due to the different speeds of

Table 1
Maximum percentage errors of the total energy over the time duration of $100T$ ($\ddot{u} + 100u(1 + 10u^2) = 0,\ u_0 = 1.5,\ v_0 = 0.0,\ T = 0.15$)

Δt	$T/10$	$T/15$	$T/20$	$T/25$	$T/50$	$T/100$	$T/200$	$T/1000$
DQ method	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Average accl. [3]	–	7.1	4.1	2.6	0.7	0.2	0.0	0.0
Central diff. [3]	–	7.7	4.3	2.7	0.7	0.2	0.0	0.0
Newmark(*) [3]	–	99.8	99.6	99.3	97.7	93.1	82.9	39.5
Wilson-θ [3]	–	93.6	87.6	79.6	36.1	7.1	1.0	0.0
Houbolt [3]	–	98.8	97.3	95.3	75.3	30.0	5.4	0.1
α-Method [3]	–	67.8	49.2	34.3	6.7	0.9	0.1	0.0
Runge–Kutta [3]	–	44.5	18.4	7.3	0.3	0.0	0.0	0.0

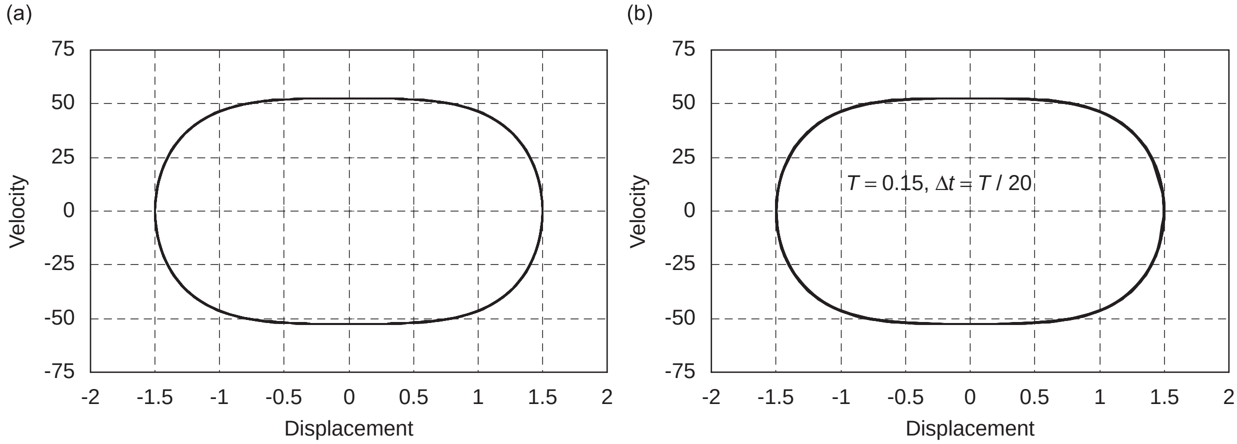

Fig. 1. Phase portraits of the hardening spring: $\ddot{u}+100u(1+10u^2)=0, u_0=1.5, v_0=0.0$. (a) Exact solution; (b) data by the DQ time integration scheme ($N=3$).

the authors' modern PCs and Xie's antique machines of the last century, since the method to solve the nonlinear equations will not affect the CPU time very much for this problem. If Newton–Raphson method is used, the re-calculated CPU time of the average acceleration method is approximately 0.041 s. It is also experienced that the skill of the programming may also affect the CPU time since it is so small for the problem considered. The total number of iteration by the DQ method and the average acceleration method is 335,009 and 376,682, respectively. Thus, the average number of iterations per time increment by the DQ method and the average acceleration method is approximately the same. It is seen that the CPU time of the DQ method is twice of that of the average acceleration method. However, the maximum percentage errors of the total energy by the DQ method (0.38×10^{-6}) is much smaller than that (0.25×10^{-2}) by the average acceleration method. It is obvious that much more accurate results are obtained by the DQ time integration scheme with the same time increment. If the time increment increases 25 times more, i.e., $\Delta t=T/40$, the CPU time and the maximum percentage errors of the total energy by the DQ method are 0.016 s and 0.84×10^{-6}. It is seen that the DQ method can still yield much more accurate results with less CPU time as compared with the average acceleration method. Thus, the advantage of the DQ method is demonstrated.

Example 2. Consider a softening elastic spring. The nonlinear dynamic equation is

$$\ddot{u}+S\tanh(u)=0, \tag{14}$$

where $S>0$. The exact total energy is

$$E=\tfrac{1}{2}\dot{u}^2+S\ln[\cosh(u)]. \tag{15}$$

For numerical data presented herein, $S=100$ and the initial conditions are $u_0=4.0$, $v_0=0.0$.

Table 2 lists the maximum percentage errors of the total energy over the time duration of $100T$. Except for the DQ results, all other data are directly cited from Ref. [3] for comparisons. Similar trends to Example 1 are observed. It is seen again that the DQ method yields the smallest of the maximum percentage errors of the total energy. However, the time increment to reach the zero maximum percentage errors of the total energy by the DQ method increases to $\Delta t=T/25$. The phase portraits of the exact solution and the DQ solution with $\Delta t=T/20$ are given in Fig. 2. It is seen that good agreement is obtained. According to Xie [3], the average acceleration method, the method of central difference, and the fourth-order Runge–Kutta method can give reasonable good results with the same time increment for this problem. Some of the results by the remaining methods in Table 2 have deviated far from the exact solutions. This can also be seen from the maximum percentage errors of the total energy listed in Table 2.

For $\Delta t=T/1000$ and the time duration of $100T$, the CPU time on a Personal Computer by the DQ method and the average acceleration method is approximately 0.203 and 0.125 s. The maximum percentage errors of

Table 2
Maximum percentage errors of the total energy over the time duration of $100T$ ($\ddot{u} + 100\,\tanh(u) = 0,\ u_0 = 4.0,\ v_0 = 0.0,\ T = 1.14$)

Δt	$T/10$	$T/15$	$T/20$	$T/25$	$T/50$	$T/100$	$T/200$	$T/1000$
DQ method	0.8	0.2	0.8	0.0	0.0	0.0	0.0	0.0
Average accl. [3]	–	8.8	4.7	3.0	0.7	0.2	0.1	0.0
Central diff. [3]	–	14.0	7.3	3.9	0.9	0.0	0.1	0.0
Newmark(*) [3]	–	100.0	100.0	100.0	100.0	100.0	98.1	32.4
Wilson-θ [3]	–	100.0	100.0	100.0	36.0	505	0.8	0.0
Houbolt [3]	–	100.0	100.0	100.0	99.9	27.7	4.1	0.1
α-Method [3]	–	97.3	59.8	33.7	5.6	0.9	0.2	0.0
Runge–Kutta [3]	–	60.8	1.0	8.3	0.3	0.0	0.0	0.0

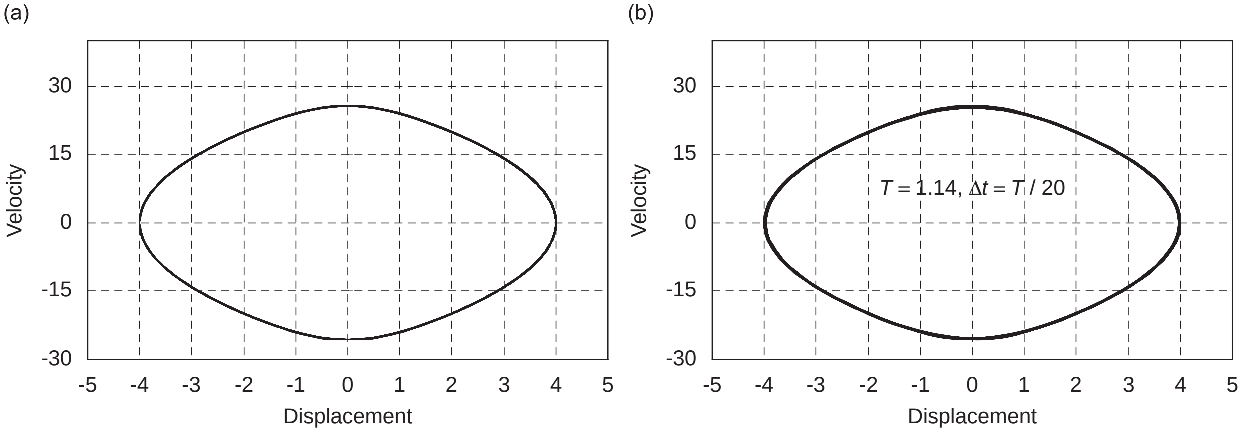

Fig. 2. Phase portraits of the softening spring: $\ddot{u} + 100\,\tanh(u) = 0, u_0 = 4.0, v_0 = 0.0$. (a) Exact solution; (b) data by the DQ time integration scheme ($N = 3$).

the total energy by the DQ method and the average acceleration method is 0.78×10^{-7} and 0.23×10^{-2}, respectively. Again much more accurate results are obtained by the DQ method. It is noted that the re-calculated CPU time for the average acceleration method (0.125 s) is much less than the one (3.72 s) given by Xie [3]. The main reason to cause the difference is also due to the different speeds of the authors' modern PCs and Xie's antique machines of the last century. The total number of iterations by the DQ method and the average acceleration method is 314,995 and 293,574, respectively. Thus, the average number of iterations per time increment by the DQ method and the average acceleration method is approximately the same. If the time increment increases 25 times more, i.e., $\Delta t = T/40$, the CPU time and the maximum percentage errors of the total energy by the DQ method are 0.031 s and 0.28×10^{-3}. Again the DQ method can yield more accurate results with less CPU time as compared with the average acceleration method if much larger time increment is used.

From Table 2, it is also observed that the percentage error in energy by the DQ method is 0.2% at $\Delta t = T/15$, but increases to 0.8% at a smaller time step ($\Delta t = T/20$). This phenomenon is similar to the Runge–Kutta method observed in Ref. [3]. Xie and Steven [6] found that a smaller time step with the average acceleration method may even change a stable solution to an unstable one. This will be further demonstrated by the following two examples.

Example 3. Consider a softening elastic spring. The nonlinear dynamic equation is given by Eq. (14) and $S = 100$. The initial conditions change to $u_0 = 10.0$, $v_0 = 0.0$.

This example was used to test the validity of the unconditional stability when the integration schemes applied to nonlinear systems. Fig. 3a–c show the results by the α-method (Fig. 3a), the DQ method with $N = 3$

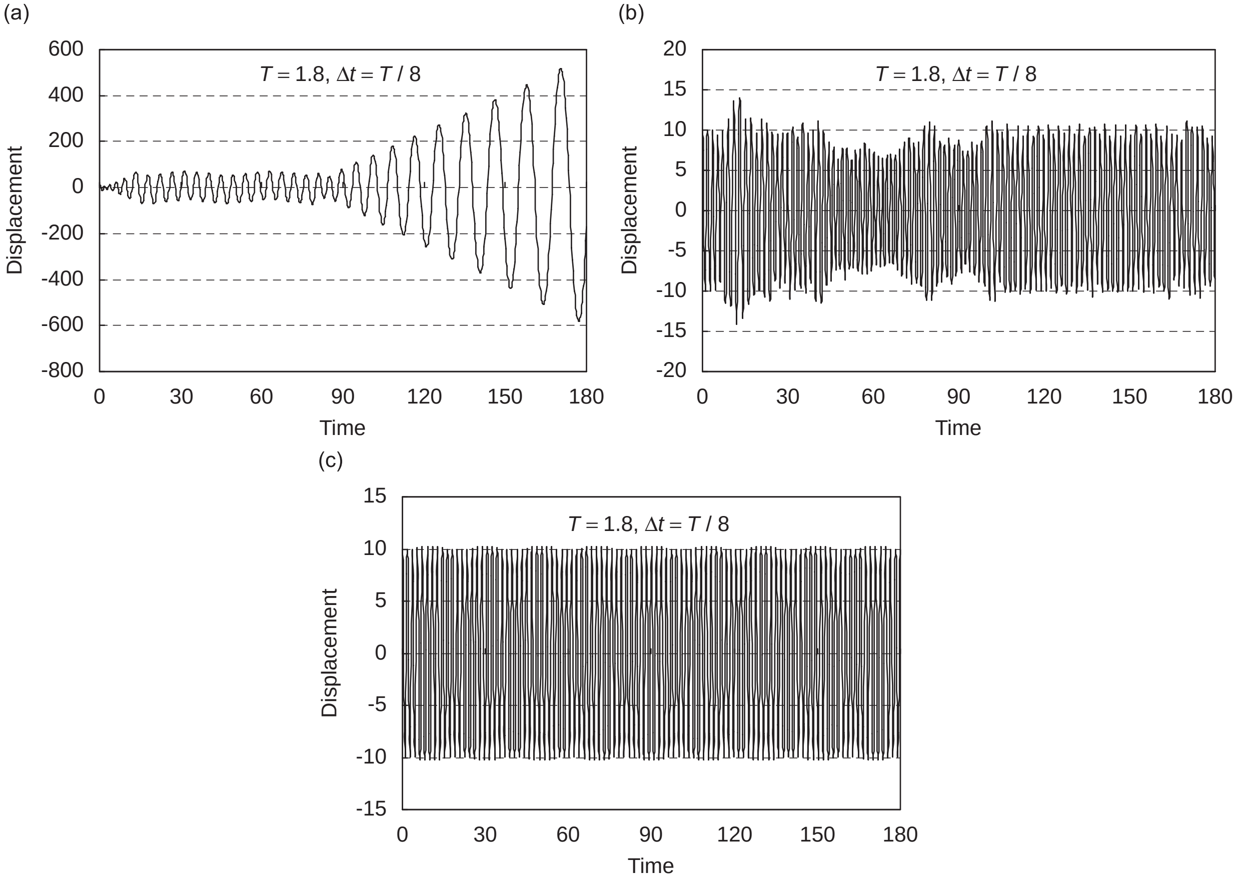

Fig. 3. Phase portraits of the softening spring: $\ddot{u} + 100\,\tanh(u) = 0, u_0 = 10.0, v_0 = 0.0$. (a) Results by the α-method; (b) data by the DQ time integration scheme ($N = 3$); and (c) data by the DQ time integration scheme ($N = 5$).

(Fig. 3b) and $N = 5$ (Fig. 3c). $\Delta t = T/8 = 1.8/8$. It is seen that the α-method, an unconditional stable time integration method in linear systems, yields unstable solutions for this case (Fig. 3a). Thus, care should be taken for choosing the time step when applying the unconditionally stable time integration schemes for nonlinear systems. The DQ method is much better than the α-method, since stable results although with some numerical damping (Fig. 3b) are obtained. The DQ method with more grid points (at the expense of increasing CPU time) in one time step can increase the accuracy (Fig. 3c).

Example 4. Consider an elastic spring. The nonlinear dynamic equation is given by Eq. (11) and $S_1 = -0.5$, $S_2 = -1$. The initial conditions are $u_0 = 0.5$, $v_0 = 0.0$.

Besides the possible instability mentioned in Example 3, other problems may arise from the numerical time integration if a large time step is used. Xie [3] demonstrated by this example that the numerical damping in the investigated schemes was so high that the results bear no resemblance to the exact solution. Figs. 4a–c show the phase portraits of the exact solution, solutions by the average acceleration method (the best one among the six time integration schemes investigated by Xie [3]) and the DQ method ($N = 3$) with $\Delta t \approx T/3 = 25$. It is experienced by the authors that the iteration process diverged for the Newmark method with $\beta = 0.0$, $\gamma = 0.5$, and Newmark method with $\beta = 0.3025$, $\gamma = 0.0$. It is seen that although the results by the DQ method ($N = 3$) are not quite accurate, but are still much better than all other methods since most results in Ref. [3] bear no resemblance to the exact solution at all. It is also found that results obtained by the average acceleration method are also sensitive to the method used in solving the nonlinear equations.

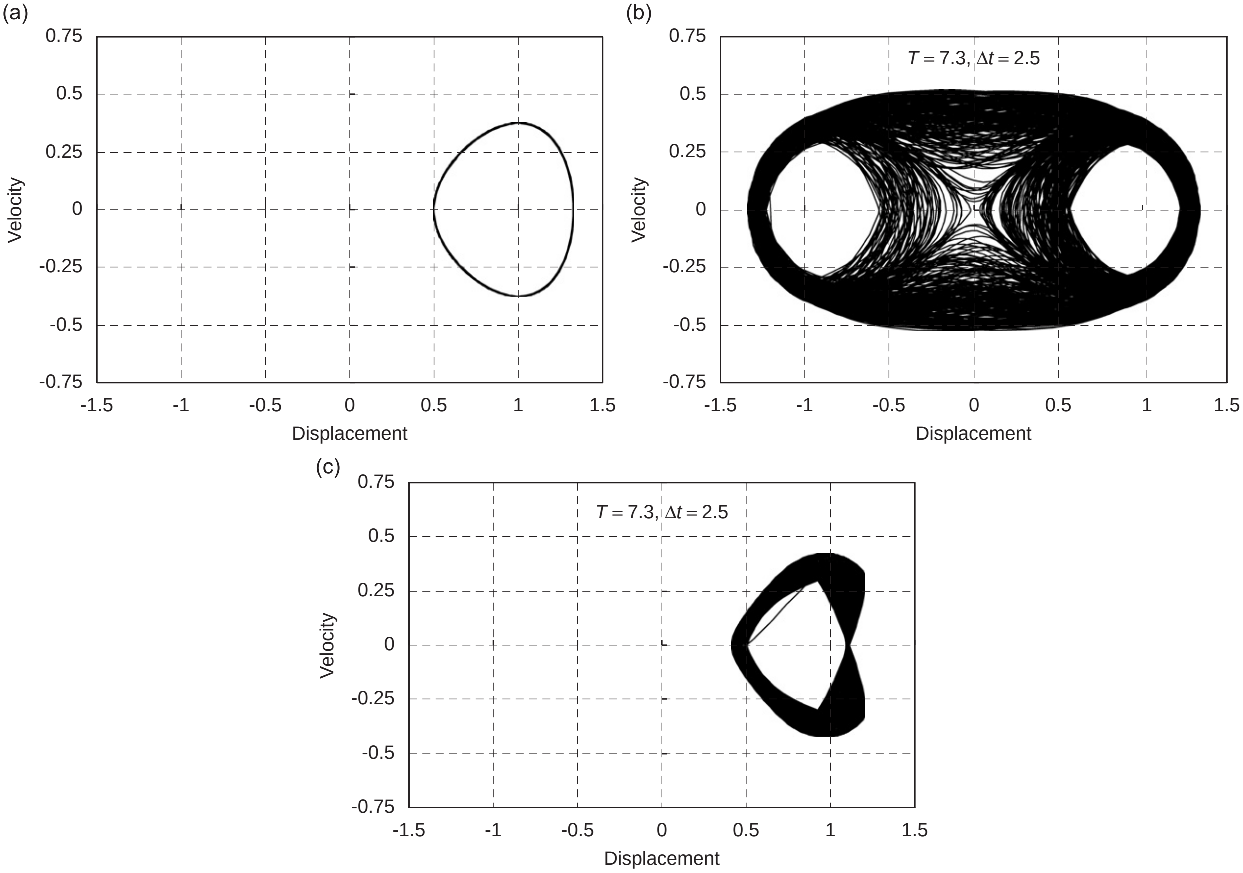

Fig. 4. Phase portraits of the softening spring: $\ddot{u} - 0.5u(1 - u^2) = 0, u_0 = 0.5, v_0 = 0.0$. (a) Exact solution; (b) average acceleration method; and (c) data by the DQ time integration scheme ($N = 3$).

4. Conclusions

Based on the numerical results reported herein, one may conclude that overall speaking the DQ time integration scheme is reliable, computationally more efficient and also suitable for time integrations over long time duration. But care should be taken in choosing a time step when applying the DQ method to nonlinear systems. Similar to all conventional unconditionally stable time integration schemes, the unconditionally stable DQ time integration scheme may also be possible to yield inaccurate results for nonlinear systems with an inappropriate too large time step.

Acknowledgment

The research was partially supported by the Natural Science Foundation of China (50675100).

References

[1] T.C. Fung, Solving initial value problem by differential quadrature method—part 1: first-order equations, *International Journal for Numerical Methods in Engineering* 50 (2001) 1411–1427.

[2] T.C. Fung, Solving initial value problem by differential quadrature method—part2: second- and-higher-order equations, *International Journal for Numerical Methods in Engineering* 50 (2001) 1429–1454.

[3] Y.M. Xie, An assessment of time integration schemes for non-linear dynamic equations, *Journal of Sound and Vibration* 192 (1) (1996) 321–331.

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

[5] Y. Wang, Differential Quadrature Method and Differential Qudrature Element Method—Principle and Applications, PhD Dissertation, Nanjing University of Aeronautics & Astronautics, Nanjing, China, 2001.

[6] Y.M. Xie, G.P. Steven, Instability, chaos, and growth and decay of energy of time-stepping schemes for non-linear dynamic equations, *Communications in Numerical Methods in Engineering* 10 (1994) 393–401.