Computational Mechanics 24 (1999) 197–205 © Springer-Verlag 1999

A Differential Quadrature as a numerical method to solve differential equations

T. Y. Wu, G. R. Liu

Abstract A Differential Quadrature proposed here can be used to solve boundary-value and initial-value differential equations with a linear or nonlinear nature. Unlike the classic Differential Quadrature Method (DQM), the newly proposed Differential Quadrature chooses the function values and some derivatives wherever necessary as independent variables. Therefore, the δ-type grid arrangement used in the classic DQM is exempt while applying the boundary conditions exactly. Most importantly, the explicit weighting coefficients can be obtained using the proposed procedures. The present method is used to solve two types of differential equations which are the single-span Bernoulli–Euler beam's buckling equation and the one-degree-of-freedom solid dynamic equation. Excellent results were obtained.

1 Introduction

The Differential Quadrature Method (DQM) was proposed by Bellman and Casti (1971) and has been employed recently in the solution of solid mechanics problems by Bert and Malik (1996a, b), Chen et al. (1997), Jang et al. (1989a, b), Kang et al. (1995), Kukreti et al. (1992) and Striz et al. (1988). Details on the development of the DQM and its application to structural mechanics problems may be found in an excellent review paper by Bert and Malik (1996a). A δ-point technique (Bert and Malik 1996a, b) has been employed in the DQM's application to boundary-value differential equations with multiple conditions. But the initial-value differential quadrature method for structural dynamics has not been reported until now to the best of the authors' knowledge.

1.1 The classic Differential Quadrature Method (DQM)

Consider a one-dimensional field variable $\psi(x)$ prescribed in a field domain $a = x_1 \le x \le x_N = b$. Let $\psi_i = \psi(x_i)$ be the function values specified in a finite set of N discrete points $x_i (i = 1, 2, \ldots, N)$ of the field domain in which the end points x_1 and x_N are included. Next, consider the value of the function derivative $\mathrm{d}^r\psi/\mathrm{d}x^r$ at some discrete points x_i, and let it be expressed as a linearly weighted sum of the function values.

$$\psi^{(r)}(x_i) = \frac{\mathrm{d}^r\psi(x_i)}{\mathrm{d}x^r} = \sum_{j=1}^{N} A_{ij}^{(r)}\psi_j \quad (i = 1, 2, \ldots, N) \tag{1}$$

where $A_{ij}^{(r)}$ are the weighting coefficients of the rth-order derivative of the function $\psi(x)$ associated with points x_i.

Equation (1), the quadrature rule for a derivative, is the essential basis of the Differential Quadrature Method. Thus using Eq. (1) for various order derivatives, one may write a given differential equation at each point of its solution domain and obtain the quadrature analog of the differential equation as a set of algebraic equations in terms of the N function values. These equations may be solved, in conjunction with the quadrature analog of the boundary conditions, to obtain the unknown function values provided that the weighting coefficients are known a priori.

The weighting coefficients may be determined by some appropriate functional approximations; and the approximate functions are referred to as test functions. The primary requirements for the choices of the test functions are of completeness in the same sense as one needs for the interpolation functions in the finite element analysis (Huebner 1975). Although there can be many choices of the test functions, a convenient and most commonly used choice in one-dimensional problems is the Lagrangian interpolation shape functions $l_j(x)$, where

$$\psi(x) = \sum_{j=1}^{N} l_j(x)\psi_j \tag{2}$$

$l_j(x)$ are the monomials of the $(N-1)$th-order polynomials. Note that the number of test functions is equal to the number of the sampling points and, for completeness, the number of sampling points should at least be equal to one plus the order of the highest derivatives.

Substituting $l_j(x)$ of Eq. (2) into Eq. (1), it may be seen that the weighting coefficients can be easily obtained. The detailed procedures can be found in references (Shu and Richards 1992, Quan and Chang 1989).

1.2 The polynomial-test-function-based weighting coefficients

The accuracy of differential quadrature solution depends on the accuracy of the weighting coefficients. To obtain accurate weighting coefficients, Quan and Chang (1989) derived explicit formulae of the Lagrangian-interpolation-

Received: 4 January 1999

T.Y. Wu (✉), G.R. Liu
Department of Mechanical and Production Engineering,
National University of Singapore,
10 Kent Ridge Crescent, Singapore, 119260

function-based weighting coefficients for the first- and second-order derivatives. Shu and Richards (1992) gave a general recurrence relationship for any high-order derivatives. These formulae were obtained by considering the test function in the Lagrangian interpolation process as in Eqs. (1) and (2). These explicit formulae's merit is that highly accurate weighting coefficients may be determined for any number of arbitrarily spaced sampling points.

But both the explicit formulae and the recursive algorithm mentioned earlier are not new. Villadsen and Michelsen (1978) and Quan and Chang (1989) have shown that the weighting coefficients of rth-order derivatives of the Lagrangian interpolation test functions are

$$A_{ij}^{(r)} = \frac{\mathrm{d}^r}{\mathrm{d}x^r} l_j(x_i) \quad (i,j = 1,2,\ldots,N) \tag{3}$$

where:

$$l_j(x) = \frac{\phi(x)}{(x-x_j)\phi^{(1)}(x_j)}; \quad \phi(x) = \prod_{m=1}^{N}(x-x_m);$$

$$\phi^{(1)}(x_j) = \frac{\mathrm{d}\phi(x_j)}{\mathrm{d}x} = \prod_{m=1;m\neq j}^{N}(x_j - x_m)$$

and x_i's are the locations of the grid points. N is the number of sampling points. Note that Eq. (3) is valid as long as linearly independent polynomials are used as the trial functions and, thus, the values of the coefficients are affected only by the distribution of the grid points. Also the linearly independent polynomials should be complete. Note that the Lagrangian interpolation shape functions $l_j(x)$ have the following properties

$$l_j(x_i) = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{4}$$

Using Eqs. (1), (2), and (3) based on Lagrangian interpolation shape functions, Quan and Chang (1989) and Shu and Richards (1992) obtained the following weighting coefficients:

$$A_{ij}^{(1)} = \frac{\mathrm{d}l_j(x_i)}{\mathrm{d}x} = \frac{\phi^{(1)}(x_i)}{(x_i - x_j)\phi^{(1)}(x_j)} \quad (i,j = 1,2,\ldots N; \quad i \neq j)$$

$$A_{ij}^{(r)} = \frac{\mathrm{d}^r l_j(x_i)}{\mathrm{d}x^r} = r\left(A_{ii}^{(r-1)}A_{ij}^{(1)} - \frac{A_{ij}^{(r-1)}}{(x_i - x_j)}\right) \quad (i,j = 1,2,\ldots N; \quad i \neq j; \quad r \geq 2)$$

$$A_{ii}^{(r)} = \frac{\mathrm{d}^r l_i(x_i)}{\mathrm{d}x^r} = -\sum_{j=1;i\neq j}^{N} A_{ij}^{(r)} (i = 1,2,\ldots,N; \quad r \geq 1) \tag{5}$$

1.3 Normal serial sampling points

A convenient and natural choice for the sampling points is that of the equally spaced sampling points. These are given in the normalized coordinate [0,1] by

$$x_i = \frac{i-1}{N-1} \quad (i = 1,2,\ldots,N) \tag{6}$$

But the Differential Quadrature solutions usually deliver more accurate results with unequally spaced sampling points. A rational basis for the sampling points is provided by the zeros of the orthogonal polynomials. A well accepted kind of sampling points in the DQM is the so-called Gauss–Lobatto–Chebyshev points:

$$x_i = \frac{1 - \cos[(i-1)\pi/(N-1)]}{2} \quad (i = 1,2,\ldots,N) \tag{7}$$

These normal serial Gauss–Lobatto–Chebyshev sampling points – Eq. (7) are used here in the beam buckling analysis.

1.4 Inverse node numbering

In the time coordinate, one usually numbers the discrete time points from the beginning accordingly. For notation's convenience in the solid dynamics problems, these authors use the inverse node numbering as in space-rocket launching. That is: one can use the initial time point as the N point and the time domain end point as the first point. These authors' numbering in dynamics problems is just contrary to the normal time-coordinate direction numbering. The suggested numbering method is very convenient for notation and programming. Please notice that in the Lagrangian interpolation process, the discrete point's numbering can be arbitrary. Thus in the normalized time coordinate τ, $\tau \subset [0,1]$, the inverse node numbering equations which correspond to Eqs. (6) and (7) in the normal serial sampling points are

$$\tau_i = \frac{N-i}{N-1} \quad (i = 1,2,\ldots,N) \tag{8}$$

$$\tau_i = \frac{1 - \cos[(N-i)\pi/(N-1)]}{2} \quad (i = 1,2,\ldots,N) \tag{9}$$

This Gauss–Lobatto–Chebyshev inverse node numbering – Eq. (9) is used here in the dynamics problems.

1.5 Recent developments of the DQM

Usually, the fourth-order differential equations in structural mechanics such as beam and plate's displacement, buckling and free-vibration analysis have two boundary equations at each boundary. Two conditions at the same point provoke a big and real challenge for the classic Differential Quadrature Method, because in the classic DQM we have only one quadrature equation at one point but two boundary equations are to be implemented. Therefore, Bert and Malik (1996a, b), Jang et al. (1989a, b), Kang et al. (1995), Kukreti et al. (1992) and Striz et al. (1988) proposed the δ-type grid arrangements, that is, besides the two boundary points, two additional adjacent points with an order of 10^{-5} (on a normalized spatial variable) distance to the boundary points were also treated as boundary points. Therefore, there are two boundary points at each boundary corresponding to their two respective boundary conditions.

In solid dynamics problems, one has two initial conditions at the initial time, that is the initial displacement and initial velocity. The same problems (two conditions at the same point) were also encountered. Therefore no one paper has appeared about solid dynamics problems solved by the classic DQM.

Although the δ-type grid arrangements work well for some circumstances, this man-made boundary grid arrangement is not mathematically sound and will sometimes cause ill-conditioned problems (Bert and Malik 1996a). No matter how small the δ distance is, it's still a domain point. A mathematically reasonable δ distance is zero. This is the essence in these authors' proposed Differential Quadrature discussed below.

Recently, Chen et al. (1997) and Wang and Gu (1997) presented a new idea about treating boundary conditions in the DQM. These improved approaches eliminate the deficiencies of the δ-type grid arrangements by applying the boundary conditions exactly. But Chen et al. (1997) used the direct linear solver for the determination of weighting coefficients. Wang and Gu (1997) only gave 3-, 4- and 5-point beam elements' explicit weighting coefficients. This paper will propose a Differential Quadrature to apply the multiple boundary or initial conditions exactly without using the δ-point technique. The explicit weighting coefficients are also presented through two types of differential equations.

2
The newly proposed Differential Quadrature

The newly proposed Differential Quadrature considers a general situation. A one-dimensional field variable $\psi(x)$ is prescribed by a differential equation in a field domain $a = x_1 \le x \le x_N = b$ and may also be constrained by a set of given conditions at any points. The solution domain is divided by points $x_i (i = 1, 2, \ldots, N)$ that include all the points with given conditions. Let $\psi_i^{(k)} = \psi^{(k)}(x_i)$ $(k = 0, 1, 2, \ldots)$ be its kth-order derivatives. Of course, $\psi_i^{(0)} = \psi_i$ are the function values. Let n_i denote the number of equations corresponding to the point x_i. The biggest k corresponding to point x_i is the number of equations (which that point has) minus one $(n_i - 1)$. These authors call $\psi_i^{(k)} = \psi^{(k)}(x_i)$ $(k = 0, 1, 2, \ldots, n_i - 1)$ the independent variables, which that point x_i has. The independent variables are chosen to be the function value and its derivatives of possible lowest order wherever necessary. For the examples discussed later, in the beam buckling analysis, $n_1 = n_N = 2$, $n_2 = n_3 = \cdots = n_{N-1} = 1$. In the dynamics problem, $n_N = 2$, $n_1 = n_2 = \cdots = n_{N-1} = 1$.

The field function's interpolation expression is constructed in the proposed Differential Quadrature just as in the numerical analysis and the finite element method. These authors use Hermite interpolation shape functions.

$$\begin{aligned}
\psi(x) &= \sum_{j=1}^{N} \Big(h_{j0}(x)\psi_j^{(0)} + h_{j1}(x)\psi_j^{(1)} + \cdots \\
&\qquad + h_{j(n_j-1)}(x)\psi_j^{(n_j-1)} \Big) \\
&= \{h_{10}(x), h_{11}(x), \ldots, h_{1(n_1-1)}(x), \ldots, \\
&\qquad h_{N0}(x), h_{N1}(x), \ldots, h_{N(n_N-1)}(x)\}^{\mathrm{T}} \\
&\qquad \times \{\psi_1^{(0)}, \psi_1^{(1)}, \ldots, \psi_1^{(n_1-1)}, \ldots, \psi_N^{(0)}, \\
&\qquad \psi_N^{(1)}, \ldots, \psi_N^{(n_N-1)}\} \\
&= \sum_{k=1}^{M} h_k(x) U_k
\end{aligned} \tag{10}$$

where

$$\begin{aligned}
M &= \sum_{j=1}^{N} n_j, \\
\{U_k\} &= \{U_1, U_2, \ldots, U_M\} \\
&= \{\psi_1^{(0)}, \psi_1^{(1)}, \ldots, . \\
&\qquad \psi_1^{(n_1-1)}, \ldots, \psi_N^{(0)}, \psi_N^{(1)} \ldots, \psi_N^{(n_N-1)}\}, \\
\{h_k\} &= \{h_1, h_2, \ldots, h_M\}^{\mathrm{T}} \\
&= \{h_{10}(x), h_{11}(x), \ldots, h_{1(n_1-1)}(x), \ldots, h_{N0}(x), \\
&\qquad h_{N1}(x), \ldots, h_{N(n_N-1)}(x)\}^{\mathrm{T}}
\end{aligned}$$

The $h_{j0}(x), \ldots, h_{j(n_j-1)}(x)$ $(j = 1, 2, \ldots, N)$ are Hermite interpolation shape functions. Their properties are listed in Table 1 when $x = x_j$, and its number's shape in Table 1 is identical to the identity matrix.

When the discrete points are not at point x_j, the values of $h_{j0}(x), \ldots, h_{j(n_j-1)}(x)$ and their derivatives of any possible order corresponding to that point are all zero. Notice that the highest order of derivative corresponding to that point is that point's total number of independent variables minus one.

Polynomial functions are often used in the FEM, but not restricted to, so are the test functions in the proposed Differential Quadrature. In this paper, polynomial functions are given as examples.

From Eq. (10) one has

Table 1. The properties of Hermite interpolation shape functions when $x = x_j$

$h_{j0}^{(i)}(x_j)$		$h_{j1}^{(i)}(x_j)$		……	$h_{j(n_j-1)}^{(i)}(x_j)$	
$h_{j0}(x_j)$	1	$h_{j1}(x_j)$	0	……	$h_{j(n_j-1)}(x_j)$	0
$h_{j0}^{(1)}(x_j)$	0	$h_{j1}^{(1)}(x_j)$	1	……	$h_{j(n_j-1)}^{(2)}(x_j)$	0
……	0	……	0	……	……	0
$h_{j0}^{(n_j-1)}(x_j)$	0	$h_{j1}^{(n_j-1)}(x_j)$	0	……	$h_{j(n_j-1)}^{(n_j-1)}(x_j)$	1

$$\frac{\mathrm{d}^r\psi(x_i)}{\mathrm{d}x^r} = \sum_{j=1}^{N}\Big(h_{j0}^{(r)}(x_i)\psi_j^{(0)} + h_{j1}^{(r)}(x_i)\psi_j^{(1)} + \cdots + h_{j(n_j-1)}^{(r)}(x_i)\psi_j^{(n_j-1)}\Big)$$
$$= \Big\{h_{10}^{(r)}(x_i), h_{11}^{(r)}(x_i), \ldots, h_{1(n_1-1)}^{(r)}(x_i), \ldots, h_{N0}^{(r)}(x_i), h_{N1}^{(r)}(x_i), \ldots, h_{N(n_N-1)}^{(r)}(x_i)\Big\}^{\mathrm{T}} \times \Big\{\psi_1^{(0)}, \psi_1^{(1)}, \ldots, \psi_1^{(n_1-1)}, \ldots, \psi_N^{(0)}, \psi_N^{(1)}, \ldots, \psi_N^{(n_N-1)}\Big\}$$
$$= \sum_{k=1}^{M} E_{ik}^{(r)} U_k \qquad (i = 1, 2, \ldots, N) \tag{11}$$

where $E_{ik}^{(r)}$ are called the weighting coefficients of the rth-order derivative of the function at point x_i.

Equation (11) is the expression of the newly proposed Differential Quadrature. Comparing the Eqs. (10) and (11), one obtains

$$E_{ik}^{(r)} = h_k^{(r)}(x_i) \tag{12}$$

The most important viewpoint in the proposed Differential Quadrature is that the total number of equations at a point is equal to the total number of the independent variables at that point, and that independent variables are always the function value and its derivatives of possible lowest order wherever necessary. In most cases, most n_j equal one, therefore only their function values are independent variables. If all n_j are one, the case is the classic Differential Quadrature Method. Therefore the classic DQM is a specific case of the proposed Differential Quadrature.

This work also gives the explicit weighting coefficients of two types of differential equations through examples. The polynomial test functions with reference to Lagrangian interpolation shape functions are used to obtain the explicit weighting coefficients. Other problems' explicit weighting coefficients can be obtained in a similar procedure.

For points with more than one equation such as single-span beam's end points, the more than one independent variable is introduced to implement the same number of equations. Then the deficiencies of the δ-type grid arrangements are eliminated, and the boundary conditions are applied directly.

The proposed Differential Quadrature can be extended to multi-dimensional problems in a similar way as the DQM. But it must be borne in mind that the total number of independent variables at a point must be equal to the total number of equations from that point. The two-dimensional problems have been discussed in another paper.

In conclusion, the DQM has only the function values as the independent variables. Therefore at one point only one differential quadrature analog can be implemented. But in the newly proposed Differential Quadrature one has the function value and its derivatives wherever necessary as the independent variables. Thus at one point, more than one differential quadrature analog can be implemented in the proposed Differential Quadrature. The resulting weighting coefficient of the DQM is a matrix of $N \times N$. But the resulting weighting coefficient of the proposed Differential Quadrature is a matrix of $N \times M$.

To illustrate the generality and accuracy of the proposed Differential Quadrature using the explicit weighting coefficients, two examples are to be cited. One example is single-span Bernoulli–Euler beam's buckling analysis, and the other example is one-degree-of-freedom dynamic analysis.

3
Single-span Bernoulli–Euler beam's buckling analysis

3.1
Differential Quadrature expression in this case

The governing equation of single-span Bernoulli–Euler beam's buckling problem is

$$\frac{\mathrm{d}^4 w}{\mathrm{d}x^4} - \frac{P}{EI}\frac{\mathrm{d}^2 w}{\mathrm{d}x^2} = 0 \qquad x \subset [0, L] = [0, 1] \tag{13}$$

where w is the transverse displacement function in the y-direction, E and I denote the modulus of elasticity and principal moment of inertia about the z-axis, respectively. P is the compressive axial load.

The single-span Bernoulli–Euler beam has four boundary conditions, two at each end. The beam is divided into $N-1$ sections according to Eq. (7). The boundary conditions are usually the following forms in the buckling analysis

$$w_i = 0; \quad w_i^{(1)} = 0; \quad EIw_i^{(2)} = 0; \quad EIw_i^{(3)} = 0 \qquad (i = 1 \text{ or } N) \tag{14}$$

In this example, one has two boundary points x_1 and x_N. At point x_1 one has two boundary conditions and thus two independent variables w_1 and $w_1^{(1)}$. At point x_N one has also two boundary conditions and then two independent variables w_N and $w_N^{(1)}$. Therefore, $n_1 = n_N = 2$, $n_2 = n_3 = \cdots = n_{N-1} = 1$, $M = \sum_{j=1}^{N} n_j = N + 2$. Now Eq. (11) adopts the following expression:

$$w^{(r)}(x_i) = \sum_{j=1}^{N} h_{j0}^{(r)}(x_i)w_j + h_{11}^{(r)}(x_i)w_1^{(1)} + h_{N1}^{(r)}(x_i)w_N^{(1)} = \sum_{j=1}^{N+2} E_{ij}^{(r)} U_j \tag{15}$$

where

$$\{E_{ij}^{(r)}\} = \{h_{10}^{(r)}(x_i), h_{11}^{(r)}(x_i), h_{20}^{(r)}(x_i), \ldots, h_{N0}^{(r)}(x_i), h_{N1}^{(r)}(x_i)\} = \{h_1^{(r)}(x_i), h_2^{(r)}(x_i), \ldots, h_{N+1}^{(r)}(x_i), h_{N+2}^{(r)}(x_i)\}$$

$$\{U_j\} = \{w_1, w_1^{(1)}, w_2, \ldots, w_N, w_N^{(1)}\} = \{U_1, U_2, \ldots, U_{N+2}\}$$

3.2
Interpolation shape function

According to the above-defined properties of the interpolation shape functions, $h_{10}(x)$ should have the following properties

$$h_{10}(x_1) = 1;\quad h_{10}^{(1)}(x_1) = 0;\quad h_{10}^{(1)}(x_N) = 0;$$
$$h_{10}(x_j) = 0, \qquad (j = 2, 3, \ldots, N) \tag{16}$$

The following $h_{10}(x)$ satisfies the fourth equation in Eq. (16)

$$h_{10}(x) = (a_1x^2 + b_1x + c_1)l_1(x) \tag{17}$$

Using the first three equations in Eq. (16), the unknown constants a_1, b_1 and c_1 in Eq. (17) can be obtained. Notice $l_1(x_N) = 0$, $l_1(x_1) = 1$ from Eq. (4), then one has

$$\begin{cases} (a_1x_1^2 + b_1x_1 + c_1)l_1(x_1) = a_1x_1^2 + b_1x_1 + c_1 = 1 \\ (2a_1x_1 + b_1)l_1(x_1) + (a_1x_1^2 + b_1x_1 + c_1)l_1^{(1)}(x_1) = 0 \\ (2a_1x_N + b_1)l_1(x_N) + (a_1x_N^2 + b_1x_N + c_1)l_1^{(1)}(x_N) = 0 \end{cases} \tag{18}$$

Notice that $l_1^{(1)}(x_N)$ can be obtained from Eq. (5) and isn't always zero. The third of Eq. (18) is satisfied only if:

$$a_1x_N^2 + b_1x_N + c_1 = 0$$

From this equation and the other two equations in Eq. (18), one obtains

$$\begin{aligned} a_1 &= \frac{-1}{(x_1 - x_N)^2} + \frac{-l_1^{(1)}(x_1)}{(x_1 - x_N)} \\ b_1 &= \frac{1}{(x_1 - x_N)} - a_1(x_1 + x_N) \\ &= \frac{2x_1}{(x_1 - x_N)^2} + \frac{(x_1 + x_N)l_1^{(1)}(x_1)}{(x_1 - x_N)} \\ c_1 &= 1 - a_1x_1^2 - b_1x_1 \\ &= \frac{(x_N - 2x_1)x_N}{(x_1 - x_N)^2} + \frac{-x_1x_Nl_1^{(1)}(x_1)}{(x_1 - x_N)} \end{aligned} \tag{19}$$

The $h_{11}(x)$ should have the following properties

$$h_{11}^{(1)}(x_1) = 1;\ h_{11}^{(1)}(x_N) = 0;$$
$$h_{11}(x_j) = 0, \quad (j = 1, 2, \ldots, N) \tag{20}$$

The following $h_{11}(x)$ satisfies the third equation in Eq. (20)

$$h_{11}(x) = (d_1x + e_1)(x - x_1)l_1(x) \tag{21}$$

Using the first two equations in Eq. (20), d_1 and e_1 can be obtained in a similar way as in Eq. (18).

$$d_1 = \frac{1}{x_1 - x_N};\quad e_1 = \frac{-x_N}{x_1 - x_N}$$

Therefore

$$\begin{aligned} h_{11}(x) &= \frac{x - x_N}{x_1 - x_N}(x - x_1)\ l_1(x) \\ &= \left(a_{11}x^2 + b_{11}x + c_{11}\right)\ l_1(x) \end{aligned} \tag{22}$$

where

$$a_{11} = \frac{1}{x_1 - x_N};\quad b_{11} = \frac{-(x_1 + x_N)}{x_1 - x_N};\quad c_{11} = \frac{x_1x_N}{x_1 - x_N}$$

The other interpolation shape functions can be obtained in an identical manner. Their properties are as follows:

$$h_{j0}(x_j) = 1;\ \ h_{j0}^{(1)}(x_1) = 0;\ h_{j0}^{(1)}(x_N) = 0;\ \ h_{j0}(x_i) = 0$$
$$(i = 1, 2, \ldots, N; j = 2, 3, \ldots, N - 1;\quad i \neq j) \tag{23}$$

$$h_{N0}(x_N) = 1;\ \ h_{N0}^{(1)}(x_N) = 0;\ \ h_{N0}^{(1)}(x_1) = 0;$$
$$h_{N0}(x_i) = 0 \qquad (i = 1, 2, \ldots, N - 1) \tag{24}$$

$$h_{N1}^{(1)}(x_N) = 1;\ \ h_{N1}^{(1)}(x_1) = 0;$$
$$h_{N1}(x_j) = 0, \qquad (j = 1, 2, \ldots, N) \tag{25}$$

Using the identical way as mentioned above, one obtains, respectively:

$$h_{j0}(x) = \left(a_jx^2 + b_jx + c_j\right)l_j(x)\ (j = 2, 3, \ldots, N - 1) \tag{26}$$

$$a_j = \frac{1}{x_j^2 - x_j(x_1 + x_N) + x_1x_N}$$
$$b_j = \frac{-(x_1 + x_N)}{x_j^2 - x_j(x_1 + x_N) + x_1x_N}$$
$$c_j = \frac{x_1x_N}{x_j^2 - x_j(x_1 + x_N) + x_1x_N}$$

$$h_{N0}(x) = \left(a_Nx^2 + b_Nx + c_N\right)l_N(x) \tag{27}$$

$$\begin{aligned} a_N &= \frac{-1}{(x_1 - x_N)^2} + \frac{l_N^{(1)}(x_N)}{(x_1 - x_N)} \\ b_N &= \frac{-1}{(x_1 - x_N)} - a_N(x_1 + x_N) \\ &= \frac{2x_N}{(x_1 - x_N)^2} - \frac{(x_1 + x_N)l_N^{(1)}(x_N)}{(x_1 - x_N)} \\ c_N &= 1 - a_Nx_N^2 - b_Nx_N \\ &= \frac{(x_1 - 2x_N)x_1}{(x_1 - x_N)^2} + \frac{x_1x_Nl_N^{(1)}(x_N)}{(x_1 - x_N)} \end{aligned}$$

$$h_{N1}(x) = \left(a_{N1}x^2 + b_{N1}x + c_{N1}\right)l_N(x) \tag{28}$$

$$a_{N1} = \frac{-1}{x_1 - x_N};\quad b_{N1} = \frac{x_1 + x_N}{x_1 - x_N};\quad c_{N1} = \frac{-x_1x_N}{x_1 - x_N}$$

3.3
Explicit weighting coefficients

From Eqs. (12), (15), (17), (22), (26), (27) and (28), one gets

$$E_{ij}^{(r)} = \frac{d^r}{dx^r}h_j(x_i)$$
$$(i = 1, 2, \ldots, N;\ j = 1, 2, \ldots, N + 2;\ r = 1, 2, 3, 4) \tag{29}$$

The above shape functions $h_j(x)$ $(j=1,2,\ldots,N+2)$ have the following form

$$F(x)=(ax^2+bx+c)l_j(x) \qquad (j=1,2,\ldots,N)$$

Thus

$$\begin{aligned}
F^{(1)}(x) &= (ax^2+bx+c)l_j^{(1)}(x)+(2ax+b)l_j(x)\\
F^{(2)}(x) &= (ax^2+bx+c)l_j^{(2)}(x)\\
&\quad +2(2ax+b)l_j^{(1)}(x)+2al_j(x)\\
F^{(3)}(x) &= (ax^2+bx+c)l_j^{(3)}(x)\\
&\quad +3(2ax+b)l_j^{(2)}(x)+6al_j^{(1)}(x)\\
F^{(4)}(x) &= (ax^2+bx+c)l_j^{(4)}(x)\\
&\quad +4(2ax+b)l_j^{(3)}(x)+12al_j^{(2)}(x)
\end{aligned} \tag{30}$$

In Eq. (30), if x is assigned a different $x_i (i=1,2,\ldots,N)$, the weighting coefficients $E_{ij}^{(r)}$ in Eq. (29) can be explicitly obtained. Notice that $l_j^{(1)}(x_i)$, $l_j^{(2)}(x_i)$, $l_j^{(3)}(x_i)$, $l_j^{(4)}(x_i)(i,j=1,2,\ldots,N)$ have been obtained in Eq. (5). Because this governing equation is a fourth-order differential equation, only the 1–4th-order weighting coefficients are needed.

Now the differential quadrature analog of the governing Eq. (13) according to Eq. (15) is

$$\sum_{j=1}^{N+2} E_{ij}^{(4)}U_j-\lambda\sum_{j=1}^{N+2} E_{ij}^{(2)}U_j=0 \qquad (i=2,3,\ldots,N-1) \tag{31}$$

where $\lambda=P/EI$. The differential quadrature analogs of the boundary condition Eq. (14) are

$$w_i=0;\ w_i^{(1)}=0;\ EI\sum_{j=1}^{N+2}E_{ij}^{(2)}U_j=0;$$

$$EI\sum_{j=1}^{N+2}E_{ij}^{(3)}U_j=0 \quad (i=1 \text{ or } N) \tag{32}$$

This work will give the normalized critical buckling load λ of the different boundary conditions such as pinned-pinned, fixed-fixed and fixed-pinned ends in Table 2. For different boundary conditions one has a proper combination of boundary Eq. (32). By rearranging Eqs. (31) and (32), the assembled form, which is similar to the expressions in reference (Bert and Malik 1996a), is

Table 2. Comparison of beam normalized critical buckling load λ under various boundary conditions

N	Pinned–Pinned	Fixed–Fixed	Fixed–Pinned
Analytic	9.869604	39.47842	20.19073
6	9.867287	40.44472	20.17477
7	9.869683	39.37706	20.18902
8	9.869631	39.48238	20.19110
9	9.869604	39.47825	20.19075
10	9.869604	39.47845	20.19072
11	9.869604	39.47842	20.19073

$$\begin{bmatrix}[S_{bb}] & [S_{bd}]\\ [S_{db}] & [S_{dd}]\end{bmatrix}\begin{Bmatrix}\{U_b\}\\ \{U_d\}\end{Bmatrix} - \lambda\begin{bmatrix}[0] & [0]\\ [Q_{db}] & [Q_{dd}]\end{bmatrix}\begin{Bmatrix}\{U_b\}\\ \{U_d\}\end{Bmatrix}=0 \tag{33}$$

where the subscripts b and d indicate the grid points used for writing the quadrature analog of the boundary conditions and the governing differential equation, respectively.

$$\{U_b\}=\{U_1,U_2,U_{N+1},U_{N+2}\}=\left\{w_1,w_1^{(1)},w_N,w_N^{(1)}\right\}.$$
$$\{U_d\}=\{U_3,U_4,\ldots,U_N\}=\{w_2,w_3,\ldots,w_{N-1}\}.$$

By matrix substructuring of Eq. (33), one has the following two equations

$$[S_{bb}]\{U_b\}+[S_{bd}]\{U_d\}=0 \tag{34}$$

$$[S_{db}]\{U_b\}+[S_{dd}]\{U_d\}-\lambda([Q_{db}]\{U_b\}+[Q_{dd}]\{U_d\})=0 \tag{35}$$

In fact, Eq. (34) is a combination of boundary condition equations, and it does not contain λ. From Eq. (34) one obtains:

$$\{U_b\}=-[S_{bb}]^{-1}[S_{bd}]\{U_d\} \tag{36}$$

Back-substituting Eq. (36) into Eq. (35), one gets

$$\left([S_{dd}]-[S_{db}][S_{bb}]^{-1}[S_{bd}]\right)\{U_d\} - \lambda\left([Q_{dd}]-[Q_{db}][S_{bb}]^{-1}[S_{bd}]\right)\{U_d\}=0 \tag{37}$$

In short notation, one obtains

$$([S]-\lambda[Q])\{U_d\}=0 \tag{38}$$

This is a generalized eigenvalue equation and can be reduced to standard eigenvalue equation. By the procedure proposed here, one obtains the normalized critical buckling axial load λ of the beam with various boundary conditions. The calculated λ is compared with analytic results in Table 2. Good agreements were obtained. When more sampling points are employed, Table 2 shows that the convergence rate is very rapid.

4 One-degree-of-freedom dynamics

4.1 The normalized dynamics equation

The simplest dynamics equation is the one-degree-of-freedom vibration equation

$$m\frac{d^2y}{dt^2}+2c\frac{dy}{dt}+ky=\overline{q}\sin(\overline{p}t) \tag{39}$$

where m is the mass, c the damping value, k the elastic coefficient. $\overline{q}$ and $\overline{p}$ are the exciting force's amplitude and frequency, respectively, and d^2y/dt^2, dy/dt and y the acceleration, velocity and displacement, respectively. One usually uses the following standard form

$$\frac{d^2y}{dt^2}+2\overline{\eta}\frac{dy}{dt}+\bar{\varpi}^2y=\frac{\overline{q}\sin(\overline{p}t)}{m} \tag{40}$$

where $\overline{\eta}$ is the damping ratio and $\bar{\omega}$ is the natural frequency.

In the differential quadrature method, one usually employs the normalized coordinate [0,1]. Here again, these authors also use the normalized coordinate [0,1]. Suppose that $\tau = t/T$, T is the time length of solution domain. Using this normalized time coordinate $\tau = t/T$, one obtains

$$\frac{d^2y}{d\tau^2} + 2\overline{\eta}T\frac{dy}{d\tau} + (\bar{\omega}T)^2 y = \frac{T^2\overline{q}\sin(\overline{p}T\tau)}{m} \tag{41}$$

where $\eta = \overline{\eta}T$ is the normalized damping ratio, $\omega = \bar{\omega}T$ the normalized natural frequency, $p = \overline{p}T$ the normalized exciting force's frequency, and $q = T^2\overline{q}/m$ the normalized exciting force's amplitude. Therefore

$$\frac{d^2y}{d\tau^2} + 2\eta\frac{dy}{d\tau} + \omega^2 y = q\sin(p\tau) \tag{42}$$

In this example the units are omitted or taken as international standard (SI) units. Besides the initial displacement $y_0 = 1$ and initial velocity $y_0^{(1)} = 1$, one has the following data

$$\bar{\omega} = 1; \quad \overline{p} = 2; \quad \overline{q}/m = 1; \quad T = \pi/2; \quad \overline{\eta} = 0.05$$

The analytic solution of Eq. (42) is

$$y = e^{-\eta\tau}(A\sin\omega_1\tau + B\cos\omega_1\tau) + C\cos p\tau + D\sin p\tau \tag{43}$$

where

$$C = \frac{-2pq\eta}{(\omega^2 - p^2)^2 + 4p^2\eta^2}; \quad D = \frac{q(\omega^2 - p^2)}{(\omega^2 - p^2)^2 + 4p^2\eta^2};$$

$$B = y_0 - C; \ \omega_1 = \sqrt{\omega^2 - \eta^2}; \ A = \frac{Ty_0^{(1)} + \eta B - Dp}{\omega_1}$$

4.2 Differential Quadrature expression in this case

In this case, the time domain $t \subset [0, T]$ is normalized to $\tau \subset [0, 1]$ and then divided into $N - 1$ sections according to Eq. (9). At the initial time point τ_N, one has two initial conditions – the initial displacement and initial velocity. Thus at point τ_N one has two independent variables y_N and $y_N^{(1)}$. Therefore, $n_N = 2$, $n_1 = n_2 = \cdots = n_{N-1} = 1$. Now Eq. (11) adopts the following expression:

$$y^{(r)}(\tau_i) = \sum_{j=1}^{N} h_{j0}^{(r)}(\tau_i)y_j + h_{N1}^{(r)}(\tau_i)y_N^{(1)} = \sum_{j=1}^{N+1} E_{ij}^{(r)}U_j \tag{44}$$

where

$$\begin{aligned}\{E_{ij}^{(r)}\} &= \{h_{10}^{(r)}(\tau_i), h_{20}^{(r)}(\tau_i), \ldots, h_{N0}^{(r)}(\tau_i), h_{N1}^{(r)}(\tau_i)\} \\ &= \{h_1^{(r)}(\tau_i), h_2^{(r)}(\tau_i), \ldots, h_{N+1}^{(r)}(\tau_i)\} \\ \{U_j\} &= \{y_1, y_2, \ldots, y_N, y_N^{(1)}\} \\ &= \{U_1, U_2, \ldots, U_N, U_{N+1}\}.\end{aligned}$$

4.3 Interpolation shape functions

According to the above-defined properties of the interpolation shape functions, $h_j(\tau)(j = 1, 2, \ldots, N-1)$ should have the following properties

$$h_j(\tau_j) = 1; \quad h_j^{(1)}(\tau_N) = 0; \quad h_j(\tau_i) = 0$$
$$(j = 1, 2, \ldots, N-1; \ i = 1, 2, \ldots, N; \ i \neq j) \tag{45}$$

$h_N(\tau)$ and $h_{N+1}(\tau)$ should have the following properties, respectively.

$$\begin{aligned}&h_N(\tau_N) = 1; \quad h_N^{(1)}(\tau_N) = 0; \\ &h_N(\tau_i) = 0 \quad (i = 1, 2, \ldots, N-1)\end{aligned} \tag{46}$$

$$\begin{aligned}&h_{N+1}(\tau_N) = 0; \quad h_{N+1}^{(1)}(\tau_N) = 1; \\ &h_{N+1}(\tau_i) = 0 \quad (i = 1, 2, \ldots, N-1)\end{aligned} \tag{47}$$

The interpolation shape functions $h_j(\tau)(j = 1, 2, \ldots, N+1)$ can be obtained in an identical manner as in the aforementioned beam problem. Here their results are given and their properties can easily be verified. Notice that $l_j(\tau)(j = 1, 2, \ldots, N)$ is defined in Eq. (3).

$$h_j(\tau) = \frac{\tau - \tau_N}{\tau_j - \tau_N} l_j(\tau) \qquad (j = 1, 2, \ldots, N-1) \tag{48}$$

$$h_N(\tau) = \left[l_N^{(1)}(\tau_N)(\tau_N - \tau) + 1\right] l_N(\tau) \tag{49}$$

$$h_{N+1}(\tau) = (\tau - \tau_N) l_N(\tau) \tag{50}$$

4.4 Explicit weighting coefficients

From Eqs. (12), (44), (48), (49) and (50), one obtains

$$E_{ij}^{(r)} = \frac{d^r}{d\tau^r} h_j(\tau_i) \quad (i = 1, 2, \ldots, N; \quad j = 1, 2, \ldots, N+1; \ r = 1, 2) \tag{51}$$

The above shape functions $h_j(\tau)(j = 1, 2, \ldots, N+1)$ are of the following form

$$F(\tau) = (a\tau + b) l_j(\tau) \quad (j = 1, 2, \ldots, N)$$

Thus

$$\begin{aligned}F^{(1)}(\tau) &= (a\tau + b) l_j^{(1)}(\tau) + a l_j(\tau) \\ F^{(2)}(\tau) &= (a\tau + b) l_j^{(2)}(\tau) + 2a l_j^{(1)}(\tau)\end{aligned} \tag{52}$$

In Eq. (52), if τ is assigned a different $\tau_i (i = 1, 2, \ldots, N)$, the weighting coefficients $E_{ij}^{(r)}$ in Eq. (51) can be explicitly obtained. Notice that $l_j^{(1)}(\tau_i)$, $l_j^{(2)}(\tau_i)$ $(i, j = 1, 2, \ldots, N)$ have been obtained in Eq. (5). Because this governing equation is a second-order differential equation, only the first- and second-order weighting coefficients are needed.

Now according to Eq. (44), the differential quadrature analog of the governing Eq. (42) is

Table 3. Comparison of displacements between the calculated results and analytic results in dynamics problems to check convergence for the first step

N	τ_i	Analytic	Relative error (%)
5	1	1.59965496	−0.720
	2	1.64322220	−0.694
	3	1.51950632	−0.681
	4	1.19930126	−0.328
7	1	1.61144389	0.116e-1
	2	1.64051469	0.114e-1
	3	1.64587199	0.111e-1
	4	1.53005910	0.885e-2
	5	1.31944193	0.620e-2
	6	1.09938225	0.262e-2
9	1	1.61125535	−0.995e-4
	2	1.62979823	−0.984e-4
	3	1.65470856	−0.965e-4
	4	1.62896462	−0.885e-4
	5	1.52992250	−0.795e-4
	6	1.37690546	−0.577e-4
	7	1.20325105	−0.340e-4
	8	1.05785891	−0.135e-4

$$\sum_{j=1}^{N+1} E_{ij}^{(2)} U_j + 2\eta \sum_{j=1}^{N+1} E_{ij}^{(1)} U_j + \omega^2 y_i = q \sin(p\tau_i) \qquad (i = 1, 2, \ldots, N-1) \quad (53)$$

The initial conditions' quadrature analogs in the normalized coordinate are

$$y_N = y(\tau = 0) = y_0; \quad y_N^{(1)} = \frac{\mathrm{d}y}{\mathrm{d}\tau}(\tau = 0) = Ty_0^{(1)} \quad (54)$$

Matrix expression of Eq. (53) is

$$[[S_\mathrm{d}]\ [S_b]]\left\{\begin{array}{c}\{y_\mathrm{d}\}\\ \{y_b\}\end{array}\right\} = \{q_\mathrm{d}\} \quad (55)$$

where

$$\{y_b\} = \left\{y_N, y_N^{(1)}\right\} = \left\{y_0, Ty_0^{(1)}\right\},$$
$$\{y_\mathrm{d}\} = \{y_1, y_2, \ldots, y_{N-1}\} = \{U_1, U_2, \ldots, U_{N-1}\}$$

By matrix substructuring, Eq. (55) is rewritten as

$$[S_\mathrm{d}]\{y_\mathrm{d}\} = \{q_\mathrm{d}\} - [S_b]\{y_b\} \quad (56)$$

Therefore, every point's displacement can be obtained from Eq. (56), and thus the velocity and acceleration can be obtained from the Differential Quadrature Eq. (44). The inverse node numbering's convenience is now seen. One need not rearrange the matrix in Eq. (55), and its substructuring is straightforward.

For the dynamics Eq. (42), many time steps are calculated to check the stability of the proposed method. The time step is $\Delta t = T$. Because one also divides the time domain in Δt, one can call them the first-order time division and the second-order time division, respectively. In the second-order division of the time domain Δt, $N = 8$ is always used in Table 4. The calculated displacements by the proposed Differential Quadrature and the analytic results are compared in Tables 3 and 4. From Table 3 one can see that the convergence is very fast, when the second-order time division N is increased. From Table 4, it's shown that the stability of this method is also very good. Note that double precission Fortran is employed in this work.

Table 4. Comparison of displacements between the calculated results and analytic results in dynamics problems to check stability

Time step	τ_i	Analytic	Relative error (%)
58	1	0.042823156	0.157e-2
	2	−0.00811146	−0.804e-2
	3	−0.14379374	−0.257e-3
	4	−0.28119012	−0.112e-3
	5	−0.29884991	0.758e-5
	6	−0.18738275	0.219e-3
	7	−0.06143679	0.769e-3
223	1	−0.0221246	0.306e-2
	2	−0.02955779	−0.219e-2
	3	0.166658201	−0.223e-3
	4	0.304214792	−0.882e-4
	5	0.319368312	0.291e-4
	6	0.203391381	0.243e-3
	7	0.073272567	0.766e-3
388	1	0.02212457	0.306e-2
	2	−0.02955780	−0.219e-2
	3	−0.1666582	−0.223e-3
	4	−0.30421477	−0.882e-4
	5	−0.31936827	0.291e-4
	6	−0.20339133	0.243e-4
	7	−0.07327251	0.766e-3

5 Conclusion

A new concept of the Differential Quadrature is proposed here. The proposed method can be extended to solve boundary-value and initial-value differential equations with a linear or nonlinear nature. Besides applying the boundary conditions exactly, the δ-type grid arrangement used in the classic DQM is exempt in the newly proposed method. Any finite boundary differential equation with finite function values, their derivatives and their combinations within its domain can be solved using the proposed method. The proposed Differential Quadrature can be extended to multi-dimensional problems in the similar way as the DQM. But the following main differences between the classic DQM and the proposed method must be borne in mind.

In the DQM one has only the function values as the independent variables. Therefore at one point only one differential quadrature analog can be implemented. But in the newly proposed Differential Quadrature one has the function values and their derivatives wherever necessary as the independent variables. The resulting weighting coefficients of the DQM is a matrix of $N \times N$. But the resulting weighting coefficients of the proposed Differential Quadrature is a matrix of $N \times M$. When $M = N$, the proposed Differential Quadrature is reduced to the classic DQM.

As examples, the Differential Quadrature expressions and their explicit weighting coefficients have been presented for the fourth-order boundary-value differential equation and the second-order initial-value differential equation. Good results were obtained in the two examples as compared with the analytic results. Most importantly, the explicit weighting coefficients of the proposed Differential Quadrature for higher-order differential equations with multiple given conditions can be obtained using this paper's procedures. These highly accurate weighting coefficients may be determined for any number of arbitrarily spaced sampling points.

References

Bellman R, Casti J (1971) Differential quadrature and long term integration. J. Math. Anal. Appl. 34:235–238

Bert CW, Malik M (1996a) Differential quadrature method in computational mechanics: a review. Appl. Mech. Rev. 49:1–27

Bert CW, Malik M (1996b) Free vibration analysis of thin cylindrical shells by the differential quadrature method. J. Pressure Vessel Technol. 118:1–12

Chen W, Striz AG, Bert CW (1997) A new approach to the differential quadrature method for fourth-order equations. Int. J. Num. Meth. Eng. 40:1941–1956

Huebner KH (1975) The finite element method for engineers. John Wiley & Sons, New York

Jang SK, Bert CW, Striz AG (1989a) Nonlinear bending analysis of orthotropic rectangular plates by the method of differential quadrature. Comput. Mech. 5:217–226

Jang SK, Bert CW, Striz AG (1989b) Application of differential quadrature to static analysis of structural components. Int. J. Num. Meth. Eng. 28:561–577

Kang K, Bert CW, Striz AG (1995) Vibration analysis of shear deformable circular arches by the differential quadrature method. J. Sound Vib. 183:353–360

Kukreti AG, Farsa J, Bert CW (1992) Fundamental frequency of tapered plates by differential quadrature. J. Eng. Mech. 118:1221–1238

Quan JR, Chang CT (1989) New insights in solving distributed system equations by the differential quadrature method – I: analysis. Comput. Chem. Eng. 13:779–788

Shu C, Richards BE (1992) Application of generalized differential quadrature to solve two-dimensional incompressible Navier-Stokes equation. Int. J. Num. Meth. Fluids 15:791–798

Striz AG, Jang SK, Bert CW (1988) Nonlinear bending analysis of the thin circular plates by differential quadrature. Thin Walled Struct. 6:51–62

Villadsen JV, Michelsen ML (1978) Solution of differential equation models by polynomial approximation. Prentice-Hall Englewood Cliffs, New Jersey

Wang X, Gu H (1997) Static analysis of frame structures by the differential quadrature element method. Int. J. Num. Meth. Eng. 40:759–788