APPLICATION OF GENERALIZED DIFFERENTIAL QUADRATURE METHOD TO STRUCTURAL PROBLEMS

H. DU, M. K. LIM AND R. M. LIN

Dynamics and Vibration Centre, School of Mechanical and Production Engineering, Nanyang Technological University, Singapore 2263

SUMMARY

This paper presents the first endeavour to exploit a generalized differential quadrature method as an accurate, efficient and simple numerical technique for structural analysis. Firstly, drawbacks existing in the method of differential quadrature (DQ) are evaluated and discussed. Then, an improved and simpler generalized differential quadrature method (GDQ) is introduced to overcome the existing drawback and to simplify the procedure for determining the weighting coefficients. Subsequently, the generalized differential quadrature is systematically employed to solve problems in structural analysis. Numerical examples have shown the superb accuracy, efficiency, convenience and the great potential of this method.

1. INTRODUCTION

Numerical approximation methods for solving partial differential equations have been widely used in various engineering fields. Classical techniques such as finite element and finite difference methods are well developed and well known. These methods can provide very accurate results by using a large number of grid points. However, in a large number of cases, reasonably approximate solutions are desired at only a few specific points in the physical domain. In order to get results even at or around a point of interest with acceptable accuracy, conventional finite element and finite difference methods still require the use of a large number of grid points. Consequently, the requirement for computer capacity is often unnecessarily large in such cases.

In seeking an alternate numerical method using fewer grid points to find results with acceptable accuracy, the method of differential quadrature (DQ) was introduced by Bellman *et al.*[1,2] The method of DQ is a global approximate method. This method is based on the ideas that the derivative of a function with respect to a co-ordinate direction can be expressed as a weighted linear sum of all the function values at all mesh points along that direction and that a continuous function can be approximated by a higher-order polynomial in the overall domain. The DQ method differs from the finite element method (FEM) in two aspects. Firstly, the FEM uses lower-order polynomials to approximate a function on a local element level, while the DQ method approximates a function on the global area using higher-order polynomials. Secondly, the DQ method directly approximates the derivatives of a function at a point, while the FEM approximates a function over a local element and the derivatives can then be derived from the approximate function. In this aspect, the DQ method is more similar to the finite difference method (FDM). However, the FDM is also a local approximation method based on lower-order polynomial approximation. In fact, it can be shown that the FDM is just a special case of the DQ method where it is applied locally on the range $[x_{i-1}, x_{i+1}]$. Owing to the higher-order

CCC 0029–5981/94/111881–16$9.00

Received 18 January1993
Revised 28 August 1993

polynomial approximation in the DQ method, it usually requires fewer grid points as compared to the FEM and FDM to achieve accurate results. However, the DQ method leads to non-symmetric and non-banded system matrices. The DQ method could be an alternative to the conventional numerical methods such as the FEM or FDM, especially when only solutions at a few specific points are required. Applications of this method to various problems showed that it has potential as an attractive approximation technique.[3-10]

However, there exists a major drawback in the originally proposed DQ. As known, the most important part of DQ is to determine the weighting coefficients for discretization of any order partial derivative. There are *two methods in use* which were proposed by Bellman *et al.*[2] to obtain the weighting coefficients. *One method* is to solve a set of linear algebraic equations which satisfy exactly the linear constrained relation for all polynomials of degree less than or equal to $N-1$. This set of equations has a unique solution because the matrix elements are composed of a Vandermonde matrix. Unfortunately, when N is large Vandermonde matrix is ill-conditioned and the inversion of this matrix becomes difficult. Moreover, a set of $N \times N$ linear algebraic equations has to be solved for each order derivative. *The other method* is to compute the weighting coefficient by an algebraic formula, but with the co-ordinates of grid points chosen as the roots of an Nth order shifted Legendre polynomial. This means that if N is specified, the distribution of grid points is fixed even for different physical problems or different boundary conditions. This creates a major drawback and restricts the application of DQ, since some practical problems may need more grids near the boundary, while some others may not. Obviously, both of the methods originally proposed for determining the weighting coefficients required by the method of DQ have a major drawback. In order to overcome these drawbacks, Shu and Richard presented a generalized differential quadrature (GDQ) and applied it to solve some fluid dynamics equations.[11,12] Preliminary results have shown that the method is very efficient and convenient.

This paper is to explore the potential of the GDQ method as an accurate, efficient and simple numerical method for structural analysis. The drawbacks existing in the original DQ method are first discussed and quantified in relation to its application to structural analysis. Then, the GDQ method is introduced. The GDQ has overcome the possible singularity problem of the original DQ in obtaining the weighting coefficients. Furthermore, in GDQ, the weighting coefficients for the first derivative and for the higher-order derivatives are given by a simple algebraic expression and a recurrence relation, respectively, and still with arbitrary choice of grid points. GDQ is finally applied to solve some practical problems in structural analysis. The ease of use and the accuracy of the GDQ are demonstrated through the numerical examples.

2. THE ORIGINAL DQ

The method of DQ is based on the idea that the partial derivative of a function with respect to a space variable at a given discrete point can be expressed as a weighted linear sum of the function values at all discrete points in the domain of that variable.

Let us take the first derivative of a one-dimensional (1-D) function $u(x, t)$ as an example, the higher-order partial derivatives will have essentially same formation. A DQ approximation of the first derivative of the function $u(x, t)$ at the ith discrete point on a grid is given by

$$u_x(x_i, t) = \sum_{j=1}^{N} c_{ij}^{(1)} u(x_j, t) \quad \text{for } i = 1, 2, \ldots, N \tag{1}$$

where $u_x(x_i, t)$ is the first derivative of $u(x, t)$ with respect to x at x_i, N is the number of discrete grids. $c_{ij}^{(1)}$ are the weighting coefficients for the first derivative approximation.

As known, the most important part of the DQ method is to determine the weighting coefficients c_{ij}. Two approaches have been used in the method of DQ.

2.1. Approach 1

The first one is to let equation (1) be exact for all polynomials of degree less than or equal to $(N-1), g(x) = x^k, k = 0, 1, \ldots, N-1$. Substituting this set of polynomials into equation (1), we have

$$\sum_{j=1}^{N} c_{ij} x_j^k = k x_i^{k-1} \quad \text{for } k = 0, 1, \ldots, N-1 \quad \text{and} \quad i = 1, 2, \ldots, N \tag{2}$$

Once the grids (i.e. x_i) are given, this relationship leads to a set of $N \times N$ linear algebraic equations. This set of equations has a unique solution since its matrix is of Vandermonde form. Unfortunately, it has been found that this set of equations become ill-conditioned and it is difficult to be solved when N is large.

In order to quantify this singularity, weighting coefficients have been calculated for equally spaced grids based on equation (2) for various number of grid points. From the computation, it is found that the maximum number of grid points is 22. Once the grid number is greater than 22, the set of the linear algebraic equations become singular and cannot be solved. The computed weighting coefficients are also compared with the accurate coefficients obtained from the generalized DQ method to be introduced later, it is found that the results from equation (2) have some errors when grid number is more than 20. Therefore, the maximum number of grid points is practically 20 for equally spaced grids if this method is used for determining the weighting coefficients. In addition, we have to solve a set of $N \times N$ linear equations for every each order of derivatives in the governing equations.

2.2. Approach 2

The other approach to determine the weighting coefficients is similar to the first one with an exception that a different set of test functions $g(x)$ is chosen for satisfying equation (1) exactly as

$$g(x) = \frac{L_N(x)}{(x - x_j) L'_N(x_j)} \quad \text{for } j = 1, 2, \ldots, N \tag{3}$$

where N is the number of the grid points. $L(x)$ is the Nth order Legendre polynomial and $L'(x)$ the first derivative of $L(x)$.

By choosing x_j to be the roots of the shifted Legendre polynomial and substituting equation (3) into equation (1), Bellman *et al.*[2] obtained a direct simple algebraic expression for the weighting coefficients $c_{ij}^{(1)}$,

$$c_{ij}^{(1)} = \frac{L'_N(x_i)}{(x_i - x_j) L'_N(x_j)} \quad \text{for } i \neq j \tag{4a}$$

$$c_{ii}^{(1)} = \frac{1 - 2x_i}{2x_i(x_i - 1)} \quad \text{for } i = j \tag{4b}$$

for $i, j = 1, 2, \ldots, N$.

It is obvious that once the number of grids N is specified, the roots of the shifted Legendre polynomial are given, thus the distribution of the grid points is fixed no matter what physical problems are considered. This imposes a major restriction on the applications of this method to problems in structural analysis, since all sorts of boundary conditions could appear and different mesh grids may be needed for different boundary conditions and different structure geometry in practice.

From the above discussion, some deficiencies are found in the originally proposed DQ. These deficiencies impose some restrictions on this method to structural analysis problems. This is probably one of the main reasons that the method of DQ is not widely used in structural analysis. In order to overcome such deficiencies, a GDQ will be introduced and applied to solve some problems in structural analysis. The ease of use and the accuracy of the GDQ will be demonstrated through the numerical examples.

3. GENERALIZED DIFFERENTIAL QUADRATURE

As described above, two approaches have been proposed for determining the weighting coefficients by previous researchers. Both of them have some drawback. The first one restricts small number of the grids to be meshed besides the need to solve sets of linear equations. The second limits the distribution of the grid points which is critical to structural analysis. To remedy these deficiencies, what we want is to find a good method to determine the weighting coefficients so that the method has no limitation on the choice of grid meshes and still gives a simple algebraic expression. Such a method was proposed by Shu and Richard[11, 12] in relation to solving some partial differential equations in fluid dynamics.

In order to find a simple algebraic expression for calculating the weighting coefficients without restricting the choice of grid meshes, let us choose the Lagrange interpolated polynomial as the set of test functions $g(x)$ instead of using the power polynomials or the Legendre polynomials:

$$g_i(x) = \frac{M(x)}{(x - x_i)M^{(1)}(x_i)} \quad \text{for } i = 1, 2, \dots, N \tag{5}$$

where

$$M(x) = \prod_{j=1}^{N} (x - x_j) \tag{6}$$

and $M^{(1)}(x)$ is the first derivative of $M(x)$ defined as,

$$M^{(1)}(x_i) = \prod_{j=1,\, j \neq i}^{N} (x_i - x_j) \tag{7}$$

and N is the number of grid points.

For simplicity, we set

$$M(x) = N(x, x_i)(x - x_i) \quad i = 1, 2, \dots, N \tag{8}$$

with

$$N(x_i, x_j) = M^{(1)}(x_i)\delta_{ij}$$

where δ_{ij} is the Kronecker operator.

Thus we have:

$$M^{(k)}(x) = N^{(k)}(x, x_j)(x - x_j) + kN^{(k-1)}(x, x_j) \quad \text{for } k = 1, 2, \dots, N - 1 \tag{9}$$

where $M^{(k)}(x)$ and $N^{(k)}(x, x_j)$ indicate the kth order derivative of $M(x)$ and $N(x, x_j)$. Substituting equation (5) into (1) yields:

$$c_{ij}^{(1)} = \frac{M^{(1)}(x_i)}{(x_i - x_j)M^{(1)}(x_j)} \quad \text{for } i \neq j \tag{10a}$$

$$c_{ii}^{(1)} = \frac{M^{(2)}(x_i)}{2M^{(1)}(x_i)} \quad \text{for } i = j \tag{10b}$$

for $i, j = 1, 2, \ldots, N$.

Equation (10) is a simple expression for computing $c_{ij}^{(1)}$ without any restriction on the choice of the co-ordinates of grid points x_i. It is obvious that once the grids (x_i) are given, $M^{(1)}(x)$ is very easy to be obtained from equation (7). Hence, $c_{ij}^{(1)}$ can be easily calculated for $i \neq j$. The calculation of $c_{ii}^{(1)}$ is based on the calculation of the second derivative of $M(x)$ which is more difficult to obtain. Instead of using equation (10a), a more convenient relationship can be obtained and used for calculating $c_{ii}^{(1)}$. It can be shown by using Taylor series expansion that the following relationship exists for $c_{ij}^{(1)}$:

$$\sum_{j=1}^{N} c_{ij}^{(1)} = 0 \quad \text{for } i = 1, 2, \ldots, N \tag{11}$$

Thus, from equation (11), the coefficient $c_{ii}^{(1)}$ can be calculated from $c_{ij}^{(1)}$ $(i \neq j)$. That is

$$c_{ii}^{(1)} = -\sum_{j=1,\, j\neq i}^{N} c_{ij}^{(1)} \quad \text{for } i = 1, 2, \ldots, N \tag{12}$$

The weighting coefficients for the second and the higher-order derivatives can be similarly obtained. Let us consider the discretization of mth order derivative of $u(x, t)$, the following DQ approximation is assumed:

$$u_x^{(m)}(x_i, t) = \sum_{j=1}^{N} c_{ij}^{(m)} u(x_j, t) \quad \text{for } i = 1, 2, \ldots, N \tag{13}$$

Again, using Lagrange interpolated polynomials as test functions, an amazing recurrence relationship will be found for the mth order weighting coefficients $c_{ij}^{(m)}$:

$$c_{ij}^{(m)} = m\left(c_{ii}^{(m-1)} c_{ij} - \frac{c_{ij}^{(m-1)}}{x_i - x_j}\right) \quad \text{for } i \neq j, \quad m = 2, 3, \ldots, N-1, \quad i, j = 1, 2, \ldots, N \tag{14}$$

where $c_{ij}^{(m)}$ is the weighting coefficients for the mth order derivative.

The calculation of $c_{ii}^{(m)}$ can be obtained from the relationship similar to equation (12):

$$c_{ii}^{(m)} = -\sum_{j=1,\, j\neq i}^{N} c_{ij}^{(m)} \quad \text{for } i = 1, 2, \ldots, N \tag{15}$$

Therefore, equations (14) and (15) together with equations (10a) and (12) give a convenient and general form for determining the weighting coefficients for the first through $(N-1)$th order derivatives. There are no restrictions on the co-ordinates of the chosen grid points. There is no need to solve for the weighting coefficients from a set of algebraic equations which could be ill-conditioned when the number of grids is large. Furthermore, this set of expressions for the determination of the weighting coefficients is so compact and simple and is very easy to be implemented in formulating and programming because of the recurrence feature. All these features give a great convenience to this GDQ for solving practical problems in structural analysis. Thus, it is of great potential to be used in structural analysis.

Extension of the method to two-dimensional (2-D) problems is straightforward. Each dimension can be simply treated individually as a 1-D case. Assuming that there are N_x grid points in the x-direction $x_1 \dots, x_{N_x}$, and N_y grid points in the y-direction $y_1 \dots, y_{N_y}$. The nth order partial derivative of $u(x, y)$ with respect to x and the mth order partial derivative of $u(x, y)$ with respect to y at x_i, y_j can be discretized as

$$u_x^{(n)}(x_i, y_j) = \sum_{k=1}^{N_x} c_{ik}^{(n)} u(x_k, y_j), \quad n = 1, \dots, N_x - 1 \tag{16a}$$

$$u_y^{(m)}(x_i, y_j) = \sum_{k=1}^{N_y} c_{jk}^{(m)} u(x_i, y_k), \quad m = 1, \dots, N_y - 1 \tag{16b}$$

for $i = 1, \dots, N_x$, $j = 1, \dots, N_y$.

As usual, this GDQ method can be used in structural analysis for solving both ordinary differential equations and partial differential equations. The application of this method for static problems will lead to a set of algebraic equations with the function values at the grid points as unknowns. While the application to time-dependent dynamic problems will result in a set of ordinary differential equations with the time-dependent function values at the grid points as unknowns. The time-dependent ordinary differential equations can then be solved by existing integration scheme. Finally, once the function values at all grids are obtained, it is very easy to determine the function values in the overall domain in terms of polynomial approximation, that is

$$u(x, y) = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} u(x_i, y_j) r_i(x) s_j(y) \tag{17}$$

where $r_i(x)$ and $s_j(y)$ are the Lagrange interpolated polynomials along the x- and y-direction, respectively.

4. APPLICATION OF GDQ TO STRUCTURAL ANALYSIS

The method of GDQ is used for analysing some static structural problems. The first three cases are for determining the static deflection behaviours of beams, circular plates and rectangular plates, respectively. The fourth and fifth cases are for the analysis of buckling behaviours of columns and rectangular plates under various boundary conditions. The formulations and programming are shown to be very straightforward and simple. The boundary conditions are easy to be implemented.

4.1. Static deflection of a beam under distributed load

The governing equation of a Bernoulli–Euler beam in bending is given by

$$EI \frac{d^4 w}{dx^4} + f(x) = 0, \quad 0 < x < L \tag{18}$$

where EI is the beam's flexural rigidity, $f(x)$ is the external distributed load, L is the length of the beam.

Normalizing the equation, we have

$$\frac{d^4 W}{dX^4} + F(x) = 0, \quad 0 < X < 1 \tag{19}$$

where $X = x/L$, $W = w/a$, $a = f_0 L/EI$, $F(x) = f(x)/f_0$.

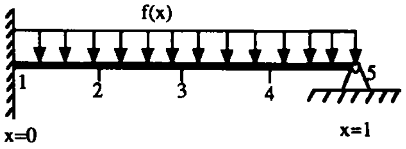

Figure 1. Grid and deflection of a slender beam

Assuming the beam is fixed at the left end and simply supported at the right end, the boundary conditions are

$$W = \frac{\mathrm{d}W}{\mathrm{d}X} = 0 \quad \text{at } X = 0 \tag{20a}$$

$$W = \frac{\mathrm{d}^2 W}{\mathrm{d}X^2} = 0 \quad \text{at } X = 1 \tag{20b}$$

Giving the number of grid points N (Figure 1) and applying the GDQ approximation to equation (19) at each discrete point on the grid, we have

$$\sum_{j=1}^{N} c_{ij}^{(4)} W_j = -F(X_i) \quad \text{for } i = 1, 2, \ldots, N \tag{21}$$

and boundary conditions:

$$W_1 = 0, \qquad \sum_{j=1}^{N} c_{1j}^{(1)} W_j = 0 \tag{22a}$$

$$W_N = 0, \qquad \sum_{j=1}^{N} c_{Nj}^{(2)} W_j = 0 \tag{22b}$$

The set of equations (21) and (22) is redundant because there are four boundary conditions (22) plus N equations in (21). In order to eliminate this redundancy, we can simply drop the equations for $i = 1, 2, (N-1)$ and N in (21). Thus.

$$\sum_{j=1}^{N} c_{ij}^{(4)} W_j = -F(X_i) \quad \text{for } i = 3, 4, \ldots, (N-2) \tag{23}$$

The combination of (23) and the boundary conditions (22) gives N equations with N unknown function values $W_1, W_2, \ldots, W_N$. The deflection of the beam under a given distributed load $f(x)$ can be obtained by solving this set of algebraic equations. It is worth pointing out that the treatment of boundary conditions here is different from that applied in solutions to structural problems using the original differential quadrature method.[10] This treatment is much simpler and easier to be implemented in programming.

Considering an uniform load with value $f(x) = f_0$, then $F(X) = 1$. The deflections of the beam at various points are presented in Table I together with the exact solutions. The exact solution of this problem is $W(X) = \frac{1}{48}X^2\,(5X - 2X^2 - 3)$. As can be seen, the numerical results are very accurate even using five grid points, i.e. $N = 5$. Up to eight digits accuracy can be achieved by using only five points. Furthermore, the formulation and programming are very simple and straightforward. The computational time is small because of the very small size of the resulting algebraic equation set (five simultaneous linear algebraic equations).

Table I. Deflection of a beam under uniformly distributed load

X	W (exact)	W (GDQ, $N = 5$)
0·0	0·0	0·0
0·25	$-0{\cdot}24414063 \times 10^{-4}$	$-0{\cdot}24414063 \times 10^{-4}$
0·5	$-0{\cdot}52083333 \times 10^{-4}$	$-0{\cdot}52083333 \times 10^{-4}$
0·75	$-0{\cdot}43945313 \times 10^{-4}$	$-0{\cdot}43945313 \times 10^{-4}$
1·0	0·0	0·0

4.2. Deflection of a thin circular plate

For a thin circular plate of uniform thickness under a general axisymmetric load, the governing equation is given as

$$\frac{\mathrm{d}^4 w}{\mathrm{d}r^4} + \frac{2}{r}\frac{\mathrm{d}^3 w}{\mathrm{d}r^3} - \frac{1}{r^2}\frac{\mathrm{d}^2 w}{\mathrm{d}r^2} + \frac{1}{r^3}\frac{\mathrm{d}w}{\mathrm{d}r} = \frac{p(r)}{D} \tag{24}$$

where D is the flexural rigidity, p is the normal pressure on the plate, r is the radial position and w the normal deflection of the plate.

Upon normalization of (24), we have

$$\frac{\mathrm{d}^4 W}{\mathrm{d}\rho^4} + \frac{2}{\rho}\frac{\mathrm{d}^3 W}{\mathrm{d}\rho^3} - \frac{1}{\rho^2}\frac{\mathrm{d}^2 W}{\mathrm{d}\rho^2} + \frac{1}{\rho^3}\frac{\mathrm{d}W}{\mathrm{d}\rho} = 1 \tag{25}$$

where $\rho = r/a$, $W = w/\alpha$, $\alpha = pa^4/D$ and a is the radius of the plate.

The regularity condition at the centre of the plate is

$$\frac{\mathrm{d}W}{\mathrm{d}\rho} = 0 \quad \text{at } \rho = 0 \tag{26}$$

The regularity condition is necessary to assure that the plate slope is zero at the origin to avoid a singularity at this location.

The boundary condition for a simply supported outside edge are

$$W = 0, \quad D\left(\frac{\mathrm{d}^2 W}{\mathrm{d}\rho^2} + \frac{v\,\mathrm{d}W}{\rho\,\mathrm{d}\rho}\right) = 0 \quad \text{at } \rho = 1 \tag{27}$$

Applying the GDQ approximation to (25) at each discrete point on the grid, we have

$$\sum_{j=1}^{N} c_{ij}^{(4)} W_j + \frac{2}{\rho_i}\sum_{j=1}^{N} c_{ij}^{(3)} W_j - \frac{1}{\rho_i^2}\sum_{j=1}^{N} c_{ij}^{(2)} W_j + \frac{1}{\rho_i^3}\sum_{j=1}^{N} c_{ij}^{(1)} W_j = 1 \quad \text{for } i = 2, 3, \ldots, N-2 \tag{28}$$

and applying the GDQ approximation to the boundary conditions:

$$\sum_{j=1}^{N} c_{ij}^{(1)} W_j = 0, \qquad W_N = 0 \tag{29a}$$

$$\sum_{j=1}^{N} c_{Nj}^{(2)} W_j + \frac{v}{\rho}\sum_{j=1}^{N} c_{Nj}^{(1)} W_j = 0 \tag{29b}$$

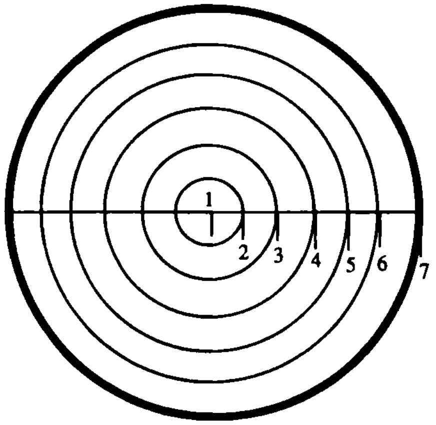

Figure 2. Grid and deflection of a circular plate

Table II. Deflection of a circular plate under uniformly distributed load using 5 grid points

ρ	W (exact)	W (GDQ, $N = 5$)
0·0	$0{\cdot}65624997 \times 10^{-1}$	$0{\cdot}65625 \times 10^{-1}$
0·25	$0{\cdot}60607907 \times 10^{-1}$	$0{\cdot}60607910 \times 10^{-1}$
0·5	$0{\cdot}46289060 \times 10^{-1}$	$0{\cdot}46289063 \times 10^{-1}$
0·75	$0{\cdot}24865721 \times 10^{-1}$	$0{\cdot}24865723 \times 10^{-1}$
1·0	0·0	0·0

Table III. Deflection of a circular plate under uniformly distributed load using 7 grid points

ρ	W (exact)	W (GDQ, $N = 7$)
0·0	$0{\cdot}65624997 \times 10^{-1}$	$0{\cdot}65625000 \times 10^{-1}$
1/6	$0{\cdot}63380109 \times 10^{-1}$	$0{\cdot}63380112 \times 10^{-1}$
2/6	$0{\cdot}56790121 \times 10^{-1}$	$0{\cdot}56790123 \times 10^{-1}$
1/2	$0{\cdot}46289060 \times 10^{-1}$	$0{\cdot}46289062 \times 10^{-1}$
4/6	$0{\cdot}32600307 \times 10^{-1}$	$0{\cdot}32600309 \times 10^{-1}$
5/6	$0{\cdot}16736592 \times 10^{-1}$	$0{\cdot}16736593 \times 10^{-1}$
1·0	0·0	0·0

Again, we only keep the discretized equations for $i = 2$ to $(N - 2)$ in (28) because there is one boundary condition at $\rho = 0$ and there are two boundary conditions at $\rho = 1$.

Solving the set of combined algebraic equations (28) and (29), the normal deflections W at various grid points can be obtained. Results are obtained by using various number of grid points at $\nu = 0{\cdot}25$. Tables II and III present the results together with the exact analytical solutions for

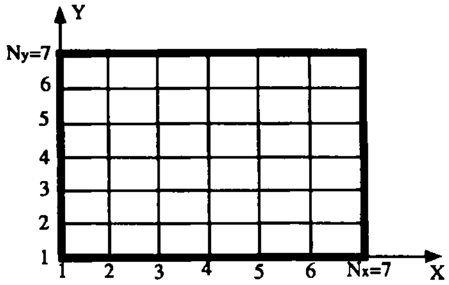

Figure 3. Grid of a rectangular plate

comparison. The exact solution can be found from the literature (e.g. Reference 13) as

$$W=\frac{1}{64}\left(\rho^4-\frac{2(3+\nu)}{(1+\nu)}\rho^2+\frac{5+\nu}{1+\nu}\right)$$

As observed, the agreement is excellent by using only a few grid points. Up to six-digits accuracy has been achieved by using only five points and seven-digits accuracy achieved using seven points. The convergence of this method is very good. Again, the size of the equations and the computational time are very small.

4.3. Deflection of rectangular plates

The governing equation for a thin rectangular plate is given as

$$\frac{\partial^4 w}{\partial x^4}+2\frac{\partial^4 w}{\partial x^2\,\partial y^2}+\frac{\partial^4 w}{\partial y^4}=p/D \tag{30}$$

where D is the flexural rigidity of the plate and p is the distributed load.

Normalizing equation (30), it becomes

$$\frac{\partial^4 W}{\partial X^4}+2\beta^2\frac{\partial^4 W}{\partial X^2\,\partial Y^2}+\beta^4\frac{\partial^4 W}{\partial Y^4}=\frac{pa^4}{D} \tag{31}$$

where $\beta=a/b$, $X=x/a$, $Y=y/b$, a is the length of the plate and b the width of the plate.

Applying the GDQ approximation (16) to (31) at each discrete point on the grid (Figure 3), we have

$$\sum_{k=1}^{N_x} c_{ik}^{(4)}W_{kj}+2\beta^2\sum_{m=1}^{N_y} c_{jm}^{(2)}\sum_{k=1}^{N_x} c_{ik}^{(2)}W_{km}+\beta^4\sum_{k=1}^{N_y} c_{jk}^{(4)}W_{ik}=\frac{pa^4}{D}$$

$$i=1,2,\ldots,N_x,\ j=1,2,\ldots,N_y \tag{32}$$

where N_x, N_y are the number of grid points along the X-direction and the Y-direction, respectively (Figure 3).

The boundary conditions for a plate clamped on all four edges are

$$W(X,0)=W(X,1)=W(0,Y)=W(1,Y)=0 \tag{33a}$$

$$\frac{\partial W}{\partial Y}(X,0)=\frac{\partial W}{\partial Y}(X,1)=\frac{\partial W}{\partial X}(0,Y)=\frac{\partial W}{\partial X}(1,Y)=0 \tag{33b}$$

Applying GDQ (16) to the boundary conditions (33)

$$W_{1j} = W_{Nj} = W_{i1} = W_{iN} = 0 \tag{34a}$$

$$\sum_{k=1}^{N_x} c_{1k}^{(1)} W_{kj} = \sum_{k=1}^{N_x} c_{Nk}^{(1)} W_{kj} = \sum_{k=1}^{N_y} c_{1k}^{(1)} W_{ik} = \sum_{k=1}^{N_y} c_{Nk}^{(1)} W_{ik} = 0 \tag{34b}$$

for $i = 1, 2, \ldots, N_x$ and $j = 2, 3, \ldots, N_y - 1$.

In order to impose the boundary conditions to equation (32), firstly, equation (34) is used to solve for W_{1j}, W_{Nj}, W_{i1}, W_{iN} (for $i = 1, 2, \ldots, N_x$ and $j = 2, 3, \ldots, N_y - 1$) in terms of the variables W_{ij} (for $i = 3, \ldots, N_x - 2$ and $j = 3, \ldots, N_y - 2$). The expressions for W_{1j}, W_{Nj}, W_{i1}, W_{iN} in terms of the variables W_{ij} (for $i = 3, \ldots, N_x - 2$ and $j = 3, \ldots, N_y - 2$) are then replaced into equation (32) to eliminate the variables W_{1j}, W_{Nj}, W_{i1}, W_{iN}, and only the discretized equations at the points $i = 3, \ldots, N_x - 2$, and $j = 3, \ldots, N_y - 2$ from (32) are to be used, Finally, the remaining set of algebraic equations can be solved to obtain the deflection of a rectangular plate. Consequently, the bending moments can also be obtained. It is worth pointing out that the coefficient matrix of the resulting set of equations is non-symmetric and non-banded.

For a plate with all four edges simply supported, the boundary conditions are

$$W(X, 0) = W(X, 1) = W(0, Y) = W(1, Y) = 0 \tag{35a}$$

$$\frac{\partial^2 W}{\partial Y^2}(X, 0) = \frac{\partial^2 W}{\partial Y^2}(X, 1) = \frac{\partial^2 W}{\partial X^2}(0, Y) = \frac{\partial^2 W}{\partial X^2}(1, Y) = 0 \tag{35b}$$

Applying GDQ (16) to the boundary conditions (35)

$$W_{1j} = W_{Nj} = W_{i1} = W_{iN} = 0 \tag{36a}$$

$$\sum_{k=1}^{N_x} c_{1k}^{(2)} W_{kj} = \sum_{k=1}^{N_x} c_{Nk}^{(2)} W_{kj} = \sum_{k=1}^{N_y} c_{1k}^{(2)} W_{ik} = \sum_{k=1}^{N_y} c_{Nk}^{(2)} W_{ik} = 0 \tag{36b}$$

for $i = 1, 2, \ldots, N_x$ and $j = 2, 3, \ldots, N_y - 1$.

Similarly, the deflection of a simply supported rectangular plate can be solved by combining the boundary conditions (36) with equation (32).

Deflections and bending moments obtained for square plates are presented in Table IV together with the exact solutions and finite element solutions available for comparison using rectangular plate element. The finite element solutions obtained by using 12 × 12 and 16 × 16 rectangular plate meshes[18] are presented in the table. For the 12 × 12 mesh, the dimension of the

Table IV. Deflection and bending moments of square plates under uniformly distributed load

Values at $x = y = a/2$	Exact	7 × 7	Error (%)	9 × 9	Error (%)	FEM[18] (12 × 12)	FEM[18] (16 × 16)
W (C–C–C–C)	0·00126	0·00123	2·3	0·00126	0·0	0·00128	0·00127
M_x (C–C–C–C)	0·0231	0·0226	2·2	0·0229	0·87	—	—
M_y (C–C–C–C)	0·0231	0·0226	2·2	0·0229	0·87	—	—
W (S–S–S–S)	0·00406	0·00396	2·5	0·00406	0·0	0·00405	0·00406
M_x (S–S–S–S)	0·0479	0·0469	2·1	0·0478	0·21	—	—
M_y (S–S–S–S)	0·0479	0·0469	2·1	0·0478	0·21	—	—

resulting system matrix is 363 × 363 for plates with four-edges clamped or 407 × 407 for plates with four-edges simply supported. For the 16 × 16 mesh, the dimension of the resulting system matrix is 588 × 588 for plates with four-edges clamped or 648 × 648 for plates with four-edges simply supported. In the analyses using the GDQ, two different sets of boundary conditions are considered. The first case is a square plate with all four-edges clamped (C–C–C–C), while the second case is a square plate with all four-edges simply supported (S–S–S–S). Results are obtained for each case using various number of grid points. It is observed that the convergence of the method is very good. Reasonably accurate results can be achieved by using only 7 × 7 grid points, for which only nine simultaneously algebraic equations need to be solved. Thus, the computational time required is tiny. Very accurate solutions can be produced with 9 × 9 grid points and for this we have to solve only a set of 25 algebraic equations. All these computations were carried out on an IBM compatible PC-486. For all of the cases, the computational time is less than 1 s.

The deflections and bending moments for rectangular plates with various width/length ratios are presented in Table V for plates with all edges clamped and in Table VI for plates with all edges simply supported, respectively. Exact results are also given together in the tables for comparison. The GDQ results are obtained using 9 × 9 grid points. As observed, both deflections and bending moments are in very good agreement with the exact solutions. Again, the computational time on an IBM compatible PC-486 is less than 1 s for all the cases.

Table V. Deflection and bending moments of rectangular plates under uniformly distributed load (C–C–C–C)

Values at $x = a/2, y = b/2$	$b/a = 1{\cdot}0$	$b/a = 1{\cdot}2$	$b/a = 1{\cdot}4$	$b/a = 1{\cdot}6$	$b/a = 1{\cdot}8$	b/$a = 2{\cdot}0$
W (exact)	0·00126	0·00172	0·00207	0·00230	0·00245	0·00254
W (GDQ)	0·00126	0·00173	0·00207	0·00230	0·00244	0·00253
Error (%)	0·00	0·58	0·00	0·00	0·41	0·39
M_x (exact)	0·0231	0·0299	0·0349	0·0381	0·0401	0·0412
M_x (GDQ)	0·0229	0·0300	0·0350	0·0382	0·0401	0·0411
Error (%)	0·87	0·33	0·29	0·26	0·00	0·24
M_y (exact)	0·0231	0·0228	0·0212	0·0193	0·0174	0·0158
M_y (GDQ)	0·0229	0·0229	0·0213	0·0193	0·0174	0·0159
Error (%)	0·87	0·44	0·47	0·00	0·00	0·63

Table VI. Deflection and bending moments of rectangular plates under uniformly distributed load (S–S–S–S)

Values at $x = a/2, y = b/2$	$b/a = 1{\cdot}0$	$b/a = 1{\cdot}2$	$b/a = 1{\cdot}4$	$b/a = 1{\cdot}6$	$b/a = 1{\cdot}8$	b/$a = 2{\cdot}0$
W (exact)	0·00406	0·00564	0·00705	0·00830	0·00931	0·0101
W (GDQ)	0·00406	0·00564	0·00707	0·00828	0·00928	0·0101
Error (%)	0·00	0·00	0·28	0·24	0·32	0·00
M_x (exact)	0·0479	0·0627	0·0755	0·0862	0·0948	0·0102
M_x (GDQ)	0·0478	0·0626	0·0754	0·0860	0·0945	0·0101
Error (%)	0·21	0·16	0·13	0·23	0·32	0·98
M_y (exact)	0·0479	0·0501	0·0502	0·0492	0·0479	0·0464
M_y (GDQ)	0·0478	0·0500	0·0502	0·0493	0·0479	0·0464
Error (%)	0·21	0·2	0·00	0·2	0·00	0·00

4.4. *Column buckling*

For the buckling behaviour of a slender elastic column, the normalized governing differential equation can be written as

$$\frac{d^2}{dX^2}\left(EI\frac{d^2W}{dX^2}\right) = -PL^2\frac{d^2W}{dX^2} \tag{37}$$

where X and W are defined in the same way as those in equation (19).

Applying the GDQ to equation (37) at each discrete point on the grid, a set of algebraic equations are obtained as

$$I\sum_{j=1}^{N} c_{ij}^{(4)} W_j + 2\frac{dI}{dX}\sum_{j=1}^{N} c_{ij}^{(3)} W_j + \frac{d^2I}{dX^2}\sum_{j=1}^{N} c_{ij}^{(2)} W_j = -\frac{PL^2}{E}\sum_{j=1}^{N} c_{ij}^{(2)} W_j \tag{38}$$

for $i = 1, 2, \ldots, N$, where I can be a given function of X.

Equation (38) can be written in a matrix form

$$[A]\{W\} = \lambda[B]\{W\} \tag{39}$$

where $\lambda = -PL^2/EI_0$, $\{W\} = [W_1, W_2, \ldots, W_N]^T$.

The buckling load can be obtained by solving the above eigenvalue problem together with appropriate boundary conditions.

Three different sets of boundary conditions are considered for both prismatic and non-prismatic columns. Using the GDQ, boundary conditions for each case can be written as

Left pinned–right pinned

$$W_1 = 0, \quad \sum_{j=1}^{N} c_{1j}^{(2)} W_j = 0 \tag{40a}$$

$$W_N = 0, \quad \sum_{j=1}^{N} c_{Nj}^{(2)} W_j = 0 \tag{40b}$$

Left fixed–right fixed

$$W_1 = 0, \quad \sum_{j=1}^{N} c_{1j}^{(1)} W_j = 0 \tag{41a}$$

$$W_N = 0, \quad \sum_{j=1}^{N} c_{Nj}^{(1)} W_j = 0 \tag{41b}$$

Left fixed–right pinned

$$W_1 = 0, \quad \sum_{j=1}^{N} c_{1j}^{(1)} W_j = 0 \tag{42a}$$

$$W_N = 0, \quad \sum_{j=1}^{N} c_{Nj}^{(2)} W_j = 0 \tag{42b}$$

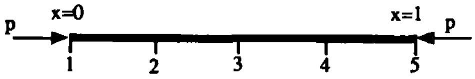

Figure 4. Grid and buckling loads of a column

The buckling loads of the columns under different boundary conditions are listed in Tables VII and VIII. Both prismatic and non-prismatic columns are considered. The exact analytical results are also given wherever available. Calculations are performed for various number of grid points. As observed, the convergence of the solution using GDQ is excellent. Comparison of the present results with the exact ones shows that the GDQ is a very accurate numerical technique. When 7 grid points are used, we only have to solve an eigenvalue problem of a 3 × 3 matrix. For 9 and 11 grid points, the eigenvalue problems of 5 × 5 and 7 × 7 matrices have to be solved, respectively. The computational time for all these cases on a PC-486 is less than 1 s.

4.5. Buckling of rectangular plates

The governing equation for the buckling of a thin rectangular plate under uniaxial load N_x is given as

$$D\frac{\partial^4 w}{\partial x^4} + 2D\frac{\partial^4 w}{\partial x^2\,\partial y^2} + D\frac{\partial^4 w}{\partial y^4} = N_x\frac{\partial^2 w}{\partial x^2} \tag{43}$$

where D is the flexural rigidity of the plate.

Normalizing equation (43), it becomes

$$\frac{\partial^4 W}{\partial X^4} + 2\beta^2\frac{\partial^4 W}{\partial X^2\,\partial Y^2} + \beta^4\frac{\partial^4 W}{\partial Y^4} = \frac{N_x a^2}{D}\frac{\partial^2 W}{\partial X^2} \tag{44}$$

where $\beta = a/b$, $X = x/a$, $Y = y/b$, a is the length of the plate and b the width of the plate.

Applying the GDQ approximation (16) to (44) at each discrete point on the grid (Figure 3), we have

$$\sum_{k=1}^{N_x} c_{ik}^{(4)} W_{kj} + 2\beta^2 \sum_{m=1}^{N_y} c_{jm}^{(2)} \sum_{k=1}^{N_x} c_{ik}^{(2)} W_{km} + \beta^4 \sum_{k=1}^{N_y} c_{jk}^{(4)} W_{ik} = \frac{N_x a^2}{D} \sum_{k=1}^{N_x} c_{ik}^{(2)} W_{kj},$$
$$i = 1, 2, \ldots, N_x, j = 1, 2, \ldots, N_y \tag{45}$$

Table VII. Buckling loads of prismatic columns

Boundary	Exact	FEM	GDQ ($N = 7$)	GDQ ($N = 9$)	GDQ ($N = 11$)
Pin–Pin	9·8696	9·9438	10·060718	9·8641905	9·8697017
Fix–Fix	39·4784	39·9730	49·090909	38·847825	39·516455
Fix–Pin	20·1421	20·4972	19·778356	20·254631	20·186532

Table VIII. Buckling loads of non-prismatic columns

Boundary	I function	Reference	Reference	GDQ ($N = 7$)	GDQ ($N = 9$)	GDQ ($N = 11$)
Pin–Pin	$(1 + X)$	15·31 [15]	14·3 [16]	14·477901	14·517996	14·511296
	$(1 + X)^2$	20·7923 [17]	27·455 [15]	19·709372	20·809884	20·804739
Fix–Fix	$(1 + X)$	—	—	56·446151	57·763035	57·345329
	$(1 + X)^2$	—	—	70·020396	84·817123	82·104358
Fix–Pin	$(1 + X)$	—	—	40·376280	29·141565	29·440638
	$(1 + X)^2$	—	—	49·294970	43·797995	41·967885

where N_x, N_y are the number of grid points along the X-direction and the Y-direction, respectively (Figure 3).

The boundary conditions for a plate clamped on all four edges (C–C–C–C) are

$$W(X,0) = W(X,1) = W(0,Y) = W(1,Y) = 0 \tag{46a}$$

$$\frac{\partial W}{\partial Y}(X,0) = \frac{\partial W}{\partial Y}(X,1) = \frac{\partial W}{\partial X}(0,Y) = \frac{\partial W}{\partial X}(1,Y) = 0 \tag{46b}$$

Applying GDQ (16) to the boundary conditions (46)

$$W_{1j} = W_{Nj} = W_{i1} = W_{iN} = 0 \tag{47a}$$

$$\sum_{k=1}^{N_x} c_{1k}^{(1)} W_{kj} = \sum_{k=1}^{N_x} c_{Nk}^{(1)} W_{kj} = \sum_{k=1}^{N_y} c_{1k}^{(1)} W_{ik} = \sum_{k=1}^{N_y} c_{Nk}^{(1)} W_{ik} = 0 \tag{47b}$$

for $i = 1, 2, \ldots, N_x$ and $j = 2, 3, \ldots, N_y - 1$.

For a plate with all four-edges simply supported (S–S–S–S), the boundary conditions are

$$W(X,0) = W(X,1) = W(0,Y) = W(1,Y) = 0 \tag{48a}$$

$$\frac{\partial^2 W}{\partial Y^2}(X,0) = \frac{\partial^2 W}{\partial Y^2}(X,1) = \frac{\partial^2 W}{\partial X^2}(0,Y) = \frac{\partial^2 W}{\partial X^2}(1,Y) = 0 \tag{48b}$$

Applying GDQ (16) to the boundary conditions (48)

$$W_{1j} = W_{Nj} = W_{i1} = W_{iN} = 0 \tag{49a}$$

$$\sum_{k=1}^{N_x} c_{1k}^{(2)} W_{kj} = \sum_{k=1}^{N_x} c_{Nk}^{(2)} W_{kj} = \sum_{k=1}^{N_y} c_{1k}^{(2)} W_{ik} = \sum_{k=1}^{N_y} c_{Nk}^{(2)} W_{ik} = 0 \tag{49b}$$

for $i = 1, 2, \ldots, N_x$ and $j = 2, 3, \ldots, N_y - 1$.

Similarly, the buckling loads of rectangular plates can be obtained by solving the eigenvalue problem in equation (45) together with appropriate boundary conditions in (47) or (49).

Buckling loads obtained for square plates are presented in Table IX together with the exact solutions. Both set of boundary conditions, four-edges clamped and four-edges simply supported, are considered. Results are obtained using various number of grid points. Good convergence of the solutions are observed from the table. For the simply supported plate, 7 × 7 grid points can produce quite accurate results. The results are obtained by solving an eigenvalue problem of a 9 × 9 matrix, for which the computational time on a PC-486 is less than 1 s. For the clamped plate, good results can be achieved by using 9 × 9 grid points. This requires a solution of an eigenvalue problem of a 25 × 25 matrix. The computational time on a PC-486 is less than 2·5 s for this case.

Table IX. Buckling loads of square plates

Boundary	Exact	7 × 7	Error (%)	9 × 9	Error (%)	11 × 11	Error (%)
S–S–S–S	39·4784	39·8652	0·98	39·4678	0·027	39·4786	0·00
C–C–C–C	99·3869	94·2733	5·1	97·1653	2·2	99·8377	0·45

5. CONCLUSIONS

Deficiencies existing in the method of DQ are evaluated and discussed. An improved and Generalized DQ method which overcomes the drawback is then introduced and applied to solve some problems in structural analysis. GDQ presents a very simple algebraic formula to determine the weighting coefficients required by the DQ approximation without restricting the choice of mesh grids. Applications of GDQ to some structural problems have shown that accurate results can be obtained using considerably few grid points, and require much less storage and computational effort. The solution procedures and the programming are much simpler and easier. And boundary conditions are easy to be incorporated in the GDQ. The superb accuracy, efficiency and convenience of this method have shown the great potential of this method for being used in structural analysis.

REFERENCES

1. R. E. Bellman and J. Casti, 'Differential quadrature and long-term integration'. *J. Math. Anal. Appl.*, **34**, 235–238 (1971).
2. R. Bellman, B. G. Kashef and J. Casti, 'Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations', *J. Comput. Phys.*, **10**, 40–52 (1972).
3. R. Bellman and B. G. Kashef, 'Solution of the partial differential equation of the Hodgkins–Huxley model using differential quadrature', *Math. Biosci.*, **19**, 1–8 (1974).
4. J. O. Mingle, 'The method of differential quadrature for transient nonlinear diffusion', *J. Math. Anal. Appl.*, **71**, 403–411 (1979).
5. G. Naadimuthu, R. E. Bellman, K. M. Wang and E. S. Lee, 'Differential quadrature and partial differential equations: some numerical results', *J. Math. Anal. Appl.*, **98**, 220–235 (1984).
6. F. Civan and C. M. Sliepcevich, 'Application of differential quadrature to transport processes', *J. Math. Anal. Appl.*, **93**, 206–221 (1983).
7. F. Civan and C. M. Sliepcevich, 'Solution of the Poisson equation by differential quadrature', *Int. j. numer. methods eng.*, **19**, 711–724 (1983).
8. C. W. Bert, S. K. Jang and A. G. Striz, 'New methods for analyzing vibration of structural components', *AIAA, J.*, **26**, 612–618 (1988).
9. A. G. Striz, S. K. Jang and C. W. Bert, 'Nonlinear bending analysis of thin circular plates by differential quadrature', *Thin-Walled struct.*, **6**, 51–62 (1988).
10. S. K. Jang, C. W. Bert and A. G. Striz, 'Application of differential quadrature to static analysis of structural components', *Int. j. numer. methods eng.*, **28**, 561–577 (1989).
11. C. Shu and B. E. Richards, 'High resolution of natural convection in a square cavity by generalized differential quadrature', *Proc. 3rd Int. Conf. on Advances in Numer. Methods in Eng.*, Theory and Applications. 11. Swansea, U.K., 1990, pp. 978–985.
12. C. Shu and B. E. Richards, 'Application of generalized differential quadrature to solve two-dimensional incompressible Navier–Stokes equations', *Int. j. numer. methods fluids*, **15**, 791–798 (1992).
13. S. Timoshenko and S. Woinowsky-Krieger, Theory of Plates and Shells, 2nd edn, McGraw-Hill, New York, 1959.
14. A. L. Newberry, C. W. Bert and A. G. Striz, 'Non-integer-polynomial finite element analysis of column buckling', *J. Struct. Eng.*, **113**, 873–878 (1987).
15. C. W. Bert, 'Improved technique for estimating buckling loads', *J. Eng. Mech.*, **110**, 1655–1665 (1984).
16. G. W. Swenson Jr., 'Analysis of nonuniform columns and beams by a simple D.C. network analyzer', *J. Aeronaut. Sci.*, **19**, 273–276 (1952).
17. F. Bleich, Buckling Strength of Metal Structures, McGraw-Hill, New York, 1952.
18. O. C. Zienkiewicz, The Finite Element Method, McGraw-Hill, New York, 1977.