ACADEMIC PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Sound and Vibration 266 (2003) 307–323

JOURNAL OF SOUND AND VIBRATION

www.elsevier.com/locate/jsvi

# A variable order approach to improve differential quadrature accuracy in dynamic analysis

Z. Zong*

*Institute of High Performance Computing, 1 Science Park Road, #01-01 The Capricorn Singapore Science Park II, Singapore 117528, Singapore*

Received 22 February 2002; accepted 16 September 2002

**Abstract**

Differential quadrature (DQ) is a numerical technique which can produce highly accurate results by using a considerably small number of grid points. When it is applied to dynamic equations, however, DQ may exhibit dynamic numerical instability. The present paper analyzed the sources of dynamic numerical instability through a simple example, and the main finding is that dynamic stability is dominated by the grid points near and on boundaries. Based on this, we propose a variable order approach which is characterized by applying different DQ schemes to the grid points near boundaries and grid points far away from boundaries. Numerical examples of both linear and non-linear dynamic equations show that the variable order approach presented in this paper may greatly improve dynamic stability, producing convincing results.

## 1. Introduction

Differential quadrature (DQ) was introduced by Bellman et al. [1] as a simple and highly efficient numerical technique in the 1970s. Its central idea is to approximate the derivatives at a point using a linear sum of function values at a set of selected grid points. As shown by Shu [2] DQ is a global method, equivalent to higher order finite-difference scheme. Compared with local numerical methods such as finite element method or low-order finite difference schemes, DQ can yield very accurate numerical results by using a considerably small number of grid points.

Since its introduction, DQ method has been applied by many authors to a variety of engineering problems [2–11]. Majority of these applications are relevant to statics or free

*Corresponding author. Tel.: +65-419-1576; fax: +65-776-1962.

*E-mail address:* zongzhi@ihpc.nus.edu.sg (Z. Zong).

doi:10.1016/S0022-460X(02)01307-X

vibrations. Recent years saw more dynamic applications [12–13]. In the dynamic cases, however, dynamic numerical instability might become a serious problem. As time marches, accumulations of numerical errors might spoil the solution. Numerical stability is an important factor when dynamic applications are concerned.

First in this paper, a simple numerical example is given to show dynamic numerical instability associated with DQ discretization. The dynamic numerical instability is then analyzed through an example. It was found that the grid points near and on boundaries have dominant influence on dynamic numerical instability. This conclusion is in fact in agreement with previous studies [3,7,8,10,14]. These studies show that there exists an optimum distribution of grid points and that more dense grid points near boundaries and lesser dense grid points in the central part of the solution domain are likely to produce better results. Based on this observation, we propose to distinguish two main classes of nodes (grid points), core nodes and cortical nodes according to their distance from boundaries. We apply variable order DQ approximations to core and cortical nodes. At core nodes, we apply higher order DQ schemes while at cortical nodes we apply lower order DQ schemes. This variable order approach turns out to be very effective in keeping dynamic stability and accuracy. Numerical examples show that the approach presented in this paper is applicable to linear and highly non-linear dynamic equations.

## 2. DQ discretization and dynamic numerical instability

Consider a continuous function $f(x,t)$ defined in terms of time $t$ and space $x$. Suppose on the following $N$ discrete spatial grid points (or nodes)

$$x_1 < x_2 < \cdots < x_N, \tag{1}$$

are defined the values of the function $f(x,t)$

$$f(x_1,t), f(x_2,t), \ldots, f(x_N,t). \tag{2}$$

It is our purpose to find the various order derivatives of $f$ with respect to $x$ at $i$th node $x_i$ using discrete function values at above $N$ grid points. DQ is characterized by approximating the derivatives of a function using a weighted sum of functions values on the grid points. The weighting coefficients are dependent on the spatial grid spacing only. Then the DQ discretization of the $r$th order partial derivative with respect to $x$ is given by the following simple equation:

$$\frac{\partial^r f(xi,t)}{\partial x^r} = \sum_{j=1}^{N} a_{ij}^r f(x_j,t), \quad i = 1,2,\ldots,N, \quad r = 1,2,\ldots,N-1, \tag{3}$$

where $a_{ij}^r$ is the weighting coefficient for the $r$th order derivative with respect to $x$. The derivatives are dependent on the function values on grid points and on spatial grid spacing.

Based on Quan and Chang [7], the explicit formulas for the first order weighting coefficients are

$$a_{ij}^1 = \frac{1}{x_i - x_j}\frac{L(x_i)}{L(x_j)}, \quad i,j = 1,\ldots,N, \ i \neq j, \tag{4a}$$

$$a^1_{ii} = -\sum_{j=1j\neq 1}^{N} a^1_{ij}, i = 1, \ldots, N. \tag{4b}$$

In Eq. (4a), the polynomial $L(x)$ is given by

$$L(x_i) = \prod_{k=1,k\neq i}^{N} (x_i - x_k), i = 1, ..., N. \tag{4c}$$

The weighting coefficients for the second and higher order derivatives are given by the following recursive equations:

$$a^r_{ij} = r\left[a^{r-1}_{ij} a^1_{ij} - \frac{a^{r-1}_{ij}}{x_i - x_j}\right], \quad i,j = 1, \ldots, N, \quad i\neq j, \quad r = 2, \ldots, N-1, \tag{5a}$$

$$a^r_{ii} = -\sum_{j=1,j\neq 1}^{N} a^r_{ij}, \quad i = 1, \ldots, N. \tag{5b}$$

These formulas are easy to be coded and can yield high accurate results. It was shown by Quan and Chang [7] that DQ is equivalent to Lagrange polynomial interpolation. If $N$ grid points are used, the interpolating polynomial is a $(N-1)$th order one. Using this idea, Shu et al. [14] obtained the approximation errors for the first order derivative $R^1$ and the second order derivative $R^2$:

$$R^1(x_i) = \frac{f^{(N)}(\xi) W^{(1)}(x_i)}{N!}, \quad i = 1, \ldots, N, \tag{6a}$$

$$R^2(x_i) = \frac{f^{(N)}(\xi) W^{(2)}(x_i)}{N!}, \quad i = 1, \ldots, N, \tag{6b}$$

where $W(x) = \prod_{j=1}^{N}(x - x_j)$ and the numbers in the superscript brackets denote the order of derivatives . These residual estimates show that very high accuracy can be obtained even if the number of grid points $N$ is not too large. Accuracy is proportional to $N$ or its powers. By 'its powers' we mean here that accuracy may also be proportional to squared or cubic $N$ or even higher order terms of $N$.

For a dynamic equation, we also need to discretize time. There have been attempts to apply DQ approach for temporal discretization, but in this paper we employ simpler temporal discretization method, fourth order Runge–Kutter (RK) scheme. Because RK method is introduced in many books, we will not explore any details of the method in the following.

To see dynamic numerical instability we first consider the following string vibration equation:

$$\frac{\partial^2 u(x,t)}{\partial t^2} = \frac{\partial^2 u(x,t)}{\partial x^2}, \quad 0\leqslant x\leqslant 1 \tag{7a}$$

subject to the following initial and boundary conditions:

$$u(x, t=0) = \sin(\pi x), \quad u(x=0,t) = u(x=1,t) = 0. \tag{7b}$$

The solution to the above equations is known to be

$$u(x,t) = \sin(\pi x)\cos(\pi t). \tag{8}$$

To apply RK method, we rewrite Eq. (7) into the following form of a set of first order ordinary differential equations:

$$\begin{cases} \dfrac{\mathrm{d}u_i}{\mathrm{d}t} = v_i, \\ \dfrac{\mathrm{d}v_i}{\mathrm{d}t} = \dfrac{\mathrm{d}^2u_i}{\mathrm{d}t^2} = \sum_{j=1}^{N} a_{ij}^2 u_i, \end{cases} \quad i = 1, \ldots, N. \tag{9}$$

In the above equations, DQ approximation was applied for the second order derivative with respect to $x$, see Eq. (3). Note that $a_{ij}^2$ in the above equation is the coefficient for the second derivative, but not squared $a_{ij}$. Eq. (9) enables us to use RK easily. For this problem we used $N = 30$ for spatial disretization. The time step used is $\Delta t = 10^{-4}$. The numerical results obtained from Eq. (9) at three time steps ($t = 0$, 50 and 100 ms) are shown in Fig. 1. The solid line denotes the displacement at $t = 50$ ms, and dotted line denote the displacement at $t = 50$ ms. From the figure, the solution becomes unstable very fast. At $t = 50$ ms, the solution exhibits divergence at both ends. The instability spreads with time to the central part of the solution domain ($x = 0$) so fast that the solution at time step $t = 100$ ms is completely meaningless. This is a very simple dynamic equation, but it strongly indicates that dynamic numerical instability may easily spoil the solution if DQ discretization of spatial variable is not carefully treated.

Both spatial and temporal discretizations might contribute to the instability. We argue that the instability does not arise from RK discretization because the Courant–Friedriches–Levy (CFL) stability condition for time step is satisfied. The CFL condition on the time step $\Delta t$ is a necessary condition for the convergence of an explicit numerical evolution algorithm [15]. Suppose $c$ is the wave velocity and $\Delta x$ is grid spacing. From Eq. (7a), the wave velocity $c = 1$ and the grid spacing $\Delta x = 1/N = 1/30 \approx 3.3 \times 10^{-2}$. Then the CFL condition

$$\Delta t = 10^{-4} \ll 3.3 \times 10^{-2} = \frac{\Delta x}{c} \tag{10}$$

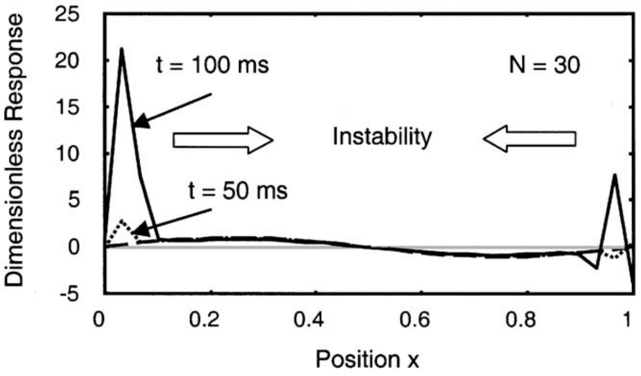


Fig. 1. Dynamic numerical instability in RK-DQ discretization of string vibration equation.

is satisfied in this example. The instability is thus solely from DQ discretization. In the following section, a simple example is given to demonstrate how numerical instability is produced.

## 3. Boundary effect and a variable order approach

To simplify the analysis, we will focus on DQ discretization only. Consider a cosine function defined on [0,2π]

$$f(x) = \cos(2x), \quad 0 \leqslant x \leqslant 2\pi. \tag{11a}$$

The second order derivative is

$$\frac{\mathrm{d}^2 f(x)}{\mathrm{d}x^2} = -4\cos(2x), \quad 0 \leqslant x \leqslant 2\pi. \tag{11b}$$

Now we use DQ to find the second order derivative of the function in Eq. (11a). Suppose $f(x)$ is approximated by its values $f_i(I = 1, \ldots, N)$ on $N$ equidistant grid points $0 = x_1 < \cdots < x_N = 2\pi$. The squared difference between the numerical value and analytical result at $i$th node is given by

$$S_i^2 = \left[\sum_{j=1}^{n} a_{ij}^2 f(x_j) + 4\cos(2x_j)\right]^2. \tag{12a}$$

The results obtained from Eq. (12a) are shown in Fig. 2(a) for $N = 20$, 30, 40, 50 and 60. The curves in the figure indicate that the errors are not uniformly distributed in the solution domain. The curves feature low accuracy near the two ends and high accuracy in the central part of the solution domain. This is not surprising at all. What is surprising is the big difference of the errors on the points near boundaries and far away from boundaries. This also agrees with our previous example (Eq. (9)) in that instability originates from the two ends and spreads fast to the rest part of the solution domain. Therefore, nodes near boundaries play the dominant role for stability.

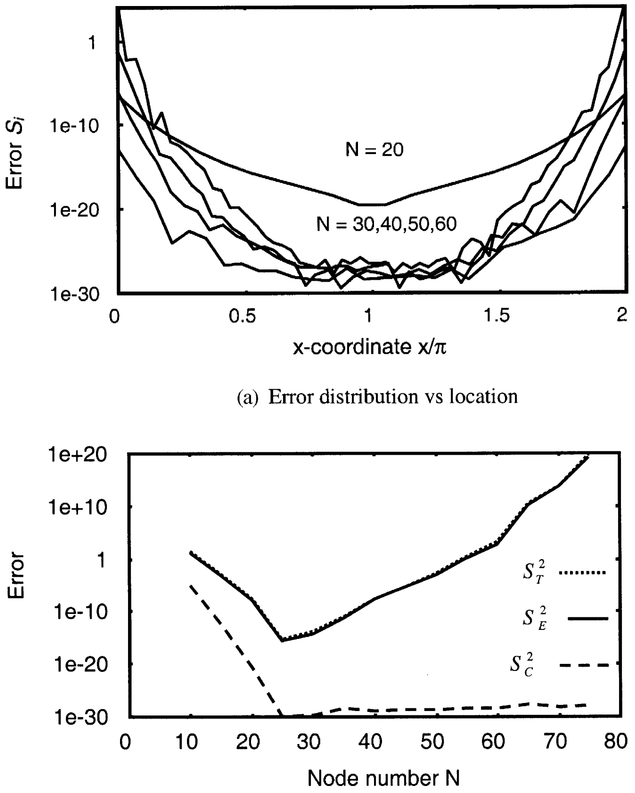
To enhance the above observations, consider the following three error measures:

$$S_T^2 = \frac{1}{N}\sum_{i=1}^{N}\left[\sum_{j=1}^{N} a_{ij}^2 f(x_j) + 4\cos(2x_j)\right]^2, \tag{12b}$$

$$S_C^2 = \left[\sum_{j=1}^{N} a_{n/2}^2 f(x_j) + 4\cos(2x_{n/2})\right]^2, \tag{12c}$$

$$S_E^2 = \left[\sum_{j=1}^{N} a_{1j}^2 f(x_j) + 4\right]^2, \tag{12d}$$

where $S_T^2$ is the average squared error, $S_C^2$ is squared error at the centre point of $x$ co-ordinate and $S_E^2$ is squared error at point $x_1$ (left end point). $S_E^2$ and $S_C^2$ are components of $S_T^2$. For different $N$'s, the error curves are shown in Fig. 2 (b). In the figure, the horizontal axis is grid number $N$, and the vertical axis is error measures in logarithmic scale. The three errors are denoted by dotted line for $S_T^2$, solid lines for $S_E^2$ and dashed line for $S_C^2$, respectively. For smaller $N$s, $S_T^2$ is large, but



(a) Error distribution vs location

(b) Error dependence on node number N

Fig. 2. Influence of node location and number on errors.

is a decreasing function of $N$. Around $N = 30$, $S_T^2$ attains minimum, and then it becomes an increasing function of the grid number $N$. Above $N = 56$, $S_T^2$ is so large that the numerical values of the derivatives are divergent. It is interesting to note that the error at the left end point $S_E^2$ is almost coincident with $S_T^2$. Contrary to $S_T^2$ and $S_E^2$, the error at the centre point $S_C^2$ is a decreasing function of grid number $N$, but above $N = 30$, $S_C^2$ is almost a constant. From the figure, we conclude that the contribution from $S_C^2$ to $S_T^2$ is negligibly small. The main contribution to the total error $S_T^2$ comes from the errors at the end points ($S_E^2$).

In summary, we conclude from the above simple example that

1. the accuracy near boundaries dominates the total accuracy;
2. the accuracy near boundaries might become very poor if grid number $N$ is large;
3. very good accuracy can be achieved at nodes far away from boundaries. The accuracy is not dependent on grid number $N$ if $N$ is large enough.

It should be noted that the above conclusions are obtained based on the example given in Eqs. (11a) and (11b). A theoretical analysis for general cases is beyond the scope of the present paper. But these conclusions are in agreement with previous studies [3,7,8,10,14]. These studies showed that a non-uniform grid distribution might produce better results when DQ discretization is applied. A frequently used grid distribution is

$$x_i = \frac{1}{2}\left[1 - \cos\left(\frac{i-1}{N-1}\pi\right)\right]^{\alpha}\Delta, \quad i = 1, \ldots, N, \tag{13}$$

where $\mathit{\Delta}$ is the interval length, and $\alpha$ is a positive constant near 1. Note that the above grid distribution is characterized by putting more nodes near boundaries and less nodes near the centre of the solution domain. For nodes near boundaries the accuracy is low, and thus dense nodes are used. Far away from boundaries, the DQ is of high accuracy and thus lesser nodes are used.

Based on the fact that boundary nodes dominate accuracy and that large grid number may lead to numerical instability, we propose to distinguish two classes of nodes, *core* and *cortical* nodes. Cortex (cortical) is an anatomical word, meaning outer shell or covering. Here we used the word to represent the nodes near and on boundaries. At a cortical node, the grid number used to approximate the derivatives at that point cannot be too large. Suppose only a small number of nodes, say $M << N$, are used to approximate the derivatives at a cortical node as shown in Fig. 3. At a core node, however, we still use $N$ nodes because the accuracy at a core node can be well kept even if grid number is large. In Fig. 3, another parameter $M_p$ is defined. It denotes transition from a core node to a cortical. Generally, we have

$$M_p << M << N. \tag{14}$$

We use **B** to denote the set of cortical nodes and **C** to denote the set of core nodes. Then the derivatives are approximated by

$$\frac{\partial^r f(x_i, t)}{\partial x^r} = \sum_{j=1}^{M} a^r_{ij} f(x_j, t), \quad x_i \in \mathbf{B}, \tag{15a}$$

$$\frac{\partial^r f(xi, t)}{\partial x^r} = \sum_{j=1}^{N} a^r_{ij} f(x_j, t), \quad xi \in \mathbf{C}. \tag{15b}$$

Or in one-dimensional case, the set B and C are

$$\mathbf{B} = \{x_1, \ldots, x_{M-Mp}\} \cup \{x_{N-M+M_P+1}, \ldots, x_N\}, \tag{16a}$$

$$\mathbf{C} = \{x_{M-M+M_{P+1}}, \ldots, x_{N-M+M_P}\}. \tag{16b}$$

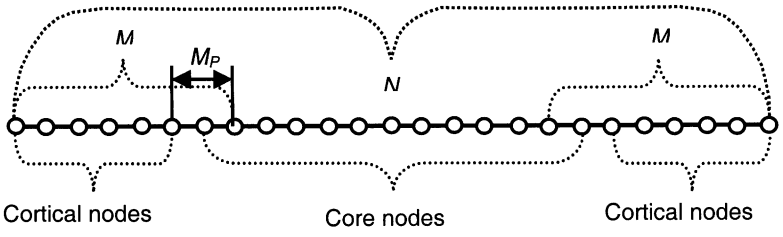


Fig. 3. Classification of cortical and core nodes.

Similar formula may be easily established for two-dimensional cases. Because DQ is equivalent to differentiation using Lagrange polynomial [2], the above equations indicate that Lagrange polynomials of different orders are separately employed to find derivatives at a core node and a cortical node. Eq. (15) is a variable order approach.

The proposed method is simple, but it does yield encouraging results. To see this, we return to the example given by Eq. (11a), the simple cosine function. Suppose $N = 55$, $M = 10$ and $M_p = 1$. The results obtained from ordinary DQ approximation and from the present scheme are shown in Fig. 4 (a) using dotted and solid lines, respectively. The analytical results are shown by dots. At both ends, the numerical results obtained from ordinary DQ are divergent while those obtained from the present method are stable at both ends. We now set $N = 70$ for which case the ordinary DQ gives very bad results as shown in Fig. 2. Using present variable order DQ, we obtained the second order derivatives at each node as shown in Fig. 4(b) denoted by solid lines. The results are both stable and in very good agreement with the analytical results.

For different node number $N$, the error defined by Eq. (12b) is shown in Fig. 5. The dotted line shows the error obtained from DQ approximation, and the solid and dashed lines show the results obtained from present method. We do see from the figure that the accuracy has been greatly improved over a wide range. The error is a decreasing function of the node number $N$ and

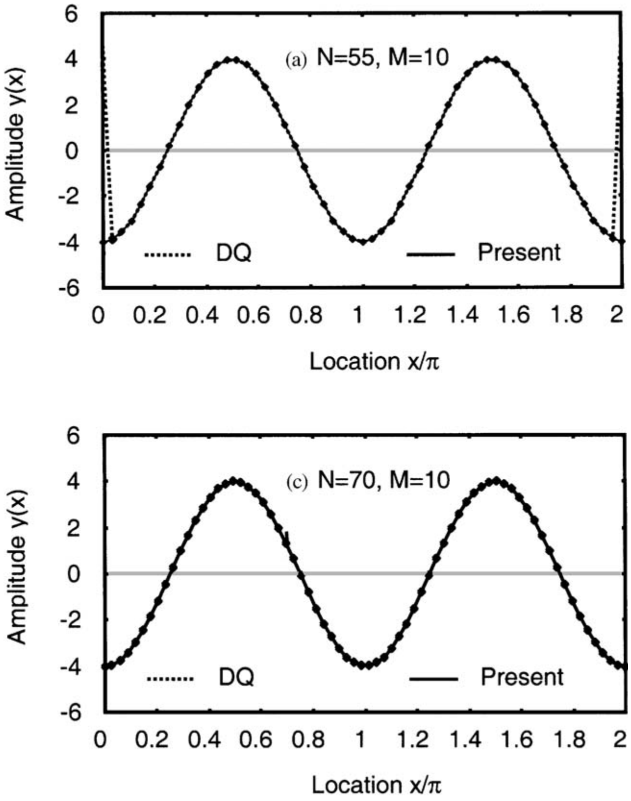


Fig. 4. Improvement by using variable order DQ schemes to cortical and core nodes.

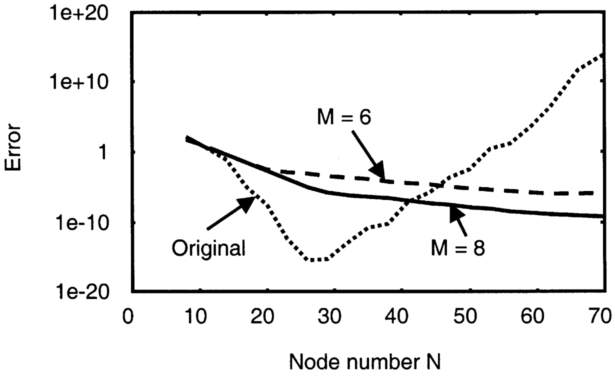


Fig. 5. Error dependence on $N$ and $M$.

accuracy does increase with $N$. Eq. (15) is only a small change to ordinary DQ, but the obtained results have been improved greatly.

Fig. 6 sows the influence of $M$ on the stability. In the figure, the results for three $M$s ($M = 3$, 4 and 6) are shown. If $M$ is too small, instability is likely to happen. Although there is a lack of theoretical value, we may obtain a rough estimate for $M$ from Figs. 5 and 6. We suggest $M = 6-15$. Then the rough estimate for $M_p$ based on Eq. (14) is $M_p = 0-5$.

## 4. Numerical examples

In this section, we present some numerical examples to show the capability of the above method.

### 4.1. String vibration

The problem is as same as the example given by Eq. (7). We used the following parameters for the computation: total node number $N = 30$ and the node number for cortical node $M = 7$. The parameter $M_p = 1$. Time step is $\Delta t = 10^{-4}$. Both simulation results (solid lines) and analytical solutions (dots) are shown in Fig. 7(a). They are in very good agreement. Recalling the example of Eq. (7), the results obtained from present method are very satisfactory.

For the sake of comparison, the general DQ with non-uniformly spaced grid points was also applied to this problem. The non-uniform grid distribution is given in Eq. (13) and the results are shown in Fig. 7(b). As expected, general DQ results are also in very good agreement with the analytical solution.

### 4.2. Scalar combustion model

The next example is a reaction–diffusion equation. This is a highly non-linear dynamic equation, in which a shock is formed. It is often employed to test suitability of a new algorithm. The equation is described in Ref. [16] as a model of a single step reaction with diffusion. The

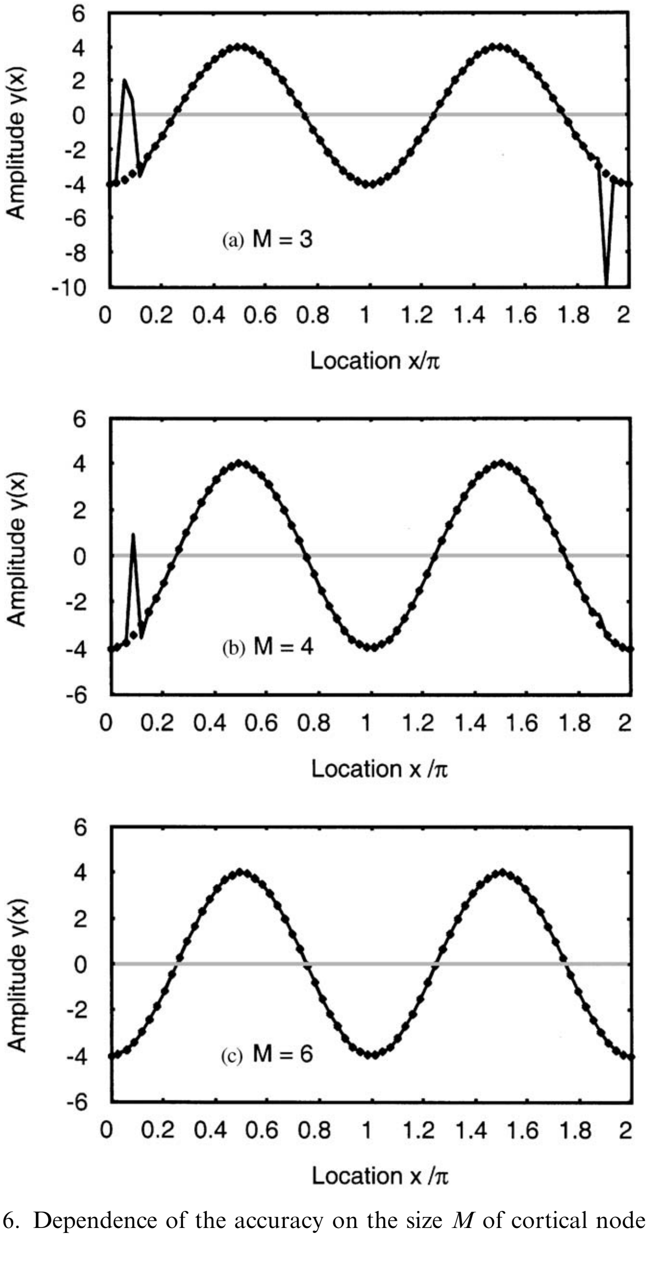


Fig. 6. Dependence of the accuracy on the size $M$ of cortical node set.

equation reads

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x^2} + D[1 + a - u(x,t)\exp(-d/u), \quad -1 \leqslant x \leqslant 1, \quad t > 0 \tag{17a}$$

subject to the following boundary and initial conditions:

$$u(-1,t) = u(1,t) = 1, \quad t > 0, \tag{17b}$$

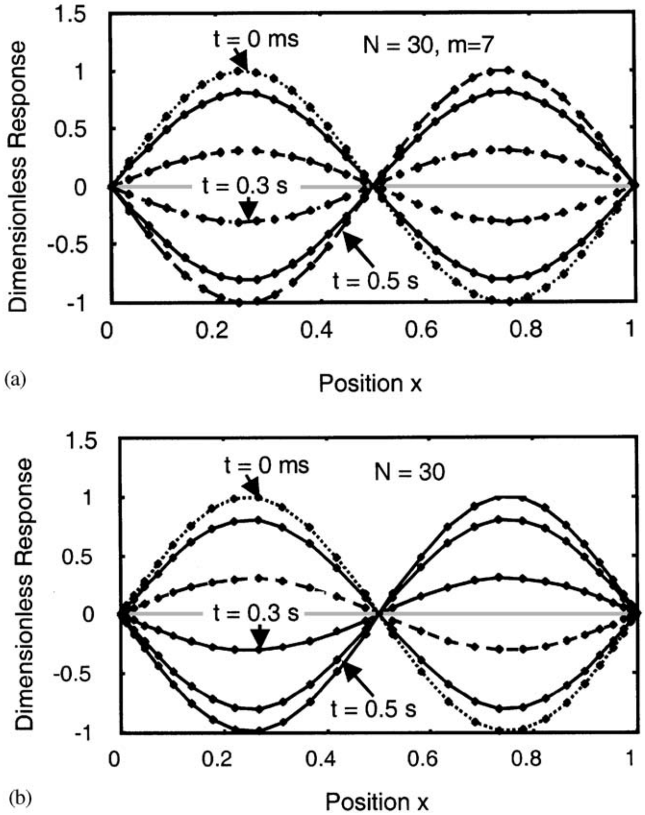


Fig. 7. String vibration using (a) present approach and general DQ(non-uniformly spaced nodes).

$$u(x,0) = 1, \quad -1 \leqslant x \leqslant 1, \tag{17c}$$

where $D = Re^d/(ad)$ and $R$, $d$, $a$ are constants. The solution represents the temperature $u(x,t)$ of a reactant in a chemical system. For small times, the temperature gradually increases from unity with a 'hot spot' forming at $x = 0$. At a finite time ignition occurs, causing the temperature at $x = 0$ to jump to $1 + a$. A sharp flame front then forms and propagates towards $x = -1$ and 1 with a speed proportional to $\exp[ad/2(1 + a)]$. In real problems, $a$ is around unity and $d$ is large; thus the flame front moves exponentially fast after ignition. The problem reaches a steady state once the flame propagates to $x = -1$ and 1.

We solved Eq. (17) for $a = 1.5$, $d = 15$ and $R = 5$ using the present variable order approach. Computed temperatures $u$ versus $x$ are shown in Fig. 8 for several time steps and $N = 30$, $M = 9$ and $M_p = 2$. The results obtained from higher order (fourth order) finite difference method [18] are also shown in the figure denoted by circles. Both are in good agreement. This is a very difficult problem due to exponential non-linearity [16]. Our method is capable of finding the solution with relative ease.

We also tried using general DQ with non-uniformly spaced grid points (the grid distribution is given in Eq. (13)) to solve Eq. (16). The results are shown in Fig. 9. In Fig. 9(a) $n = 24$ nodes were

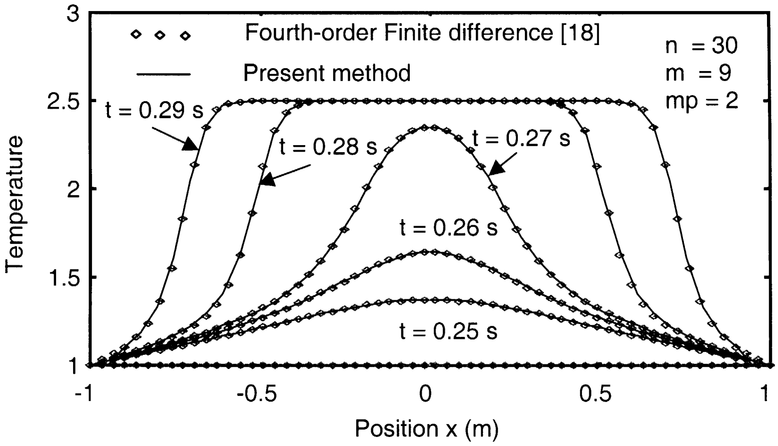


Fig. 8. Temperature vs position x at $t = 0$, 0.25, 0.26,0.27,0.28 and 0.29 for example 2, Eq. (17). Solid lines denote the results from present method while circles are results obtained from higher-order finite difference method [18].

used and in Fig. 9(b) $n = 25$ nodes were used. Insufficient accuracy is observable from Fig. 9(a), particularly at the centre ($x = 0$) at time step $t = 0.27$ s and at the flame front at $t = 0.28$ and 0.29 s. Accuracy cannot be increased by using larger number of grid points because instability occurs once node number $n$ is greater than 24. This is shown in Fig. 9(b), where 25 nodes were used and the solution became divergent as early as at time step $t = 0.027$ s.

This example shows that we can do little to improve the accuracy even if we use non-uniformly spaced grid points.

### 4.3. Two-dimensional scalar combustion model

The two-dimensional scalar reaction model is described by the following equation:

$$\frac{\partial u(x,y,t)}{\partial t} = \frac{\partial^2 u(x,y,t)}{\partial x^2} + \frac{\partial^2 u(x,y,t)}{\partial y^2} + D[1 + a - u(x,y,t)]\exp(-d/u),$$
$$-1 \leqslant x \leqslant 1, \quad -1 \leqslant y \leqslant 1 \; t > 0 \qquad (18a)$$

subject to the following boundary and initial conditions:

$$u(-1,y,t) = u(1,y,t) = 1, \quad u(x,1,t) = 1, \quad t > 0 \qquad (18b)$$

$$u(x,y,0) = 1, \quad -1 \leqslant x \leqslant 1, \quad -1 \leqslant y \leqslant 1. \qquad (18c)$$

The values used in the computation are $a = 1$, $d = 20$ and $R = 5$. It is straightforward to generalize Eq. (15) to two-dimensional cases. Suppose the two-dimensional solution domain ($a \leqslant x \leqslant b, c \leqslant x \leqslant d$) are discretized by $N_x \times N_y$ regular nodes. Applying Eq. (15) to $x$ and $y$, respectively, will yield partial derivatives in both directions. We solved Eq. (17) for $N_x = N_y = 29$, $M = 9$ and $M_p = 2$. The computed temperatures versus $(x, y)$ are shown in Fig. 10 for several time steps. We see a sharp jump of temperature from $t = 0.28$ to 0.29 s that is similar to one-

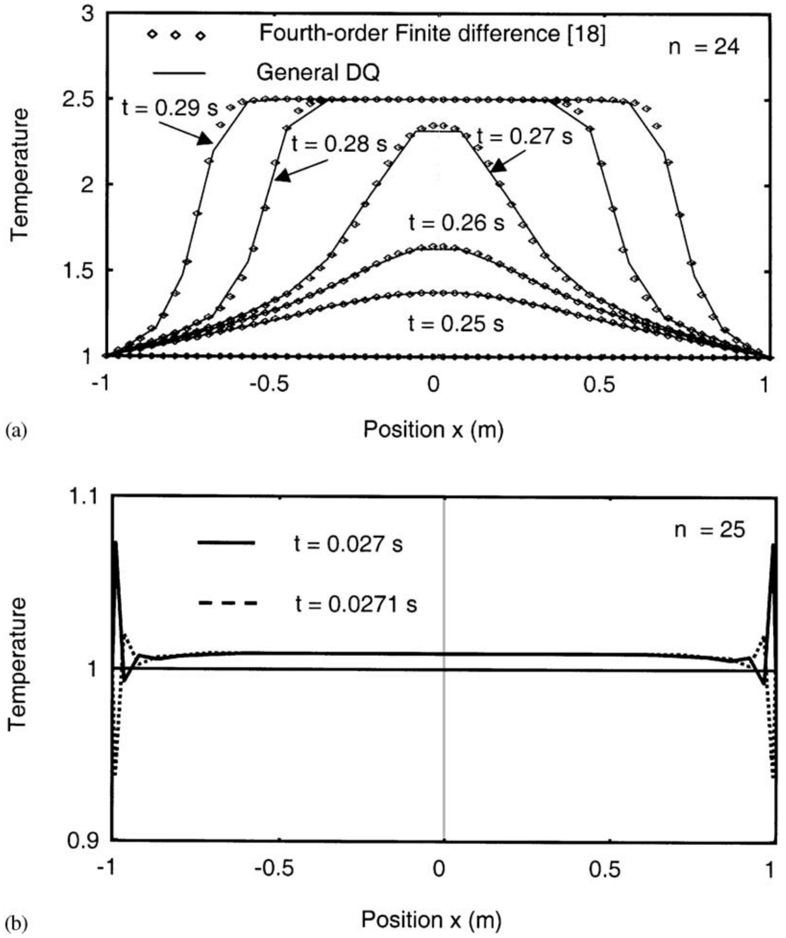


Fig. 9. The scalar combustion model solved using general DQ with non-equally spaced grid points. The node number used are (a) $n = 24$ and (b) $n = 25$. Insufficient accuracy in the first case (a) and instability in the second case (b) are observable.

dimensional case. A sharp flame front is formed at $t = 0.3\,\mathrm{s}$. The final state is shown in Fig. 10(e). The temperature in the whole domain is uniform, equal to $1 + a$. In this example, we deliberately used $N_x = N_y = 29$ just for the sake of testing the flexibility of the code.

### 4.4. Forced vibration of a simply supported plate

In all the examples previously considered, the order of partial differential equations was not higher than two. In this example, a higher order (fourth order) partial differential equation is considered to show the applicability of the current approach. Consider the forced transverse vibration of a square thin plate. The governing equation is

$$\frac{\partial^2 W}{\partial t^2} + \frac{D}{m}\nabla^2\nabla^2 W(x,y;t) = \frac{f(x,y;t)}{m} = g(t)\sin(2\pi x)\sin(\pi y), \tag{19a}$$

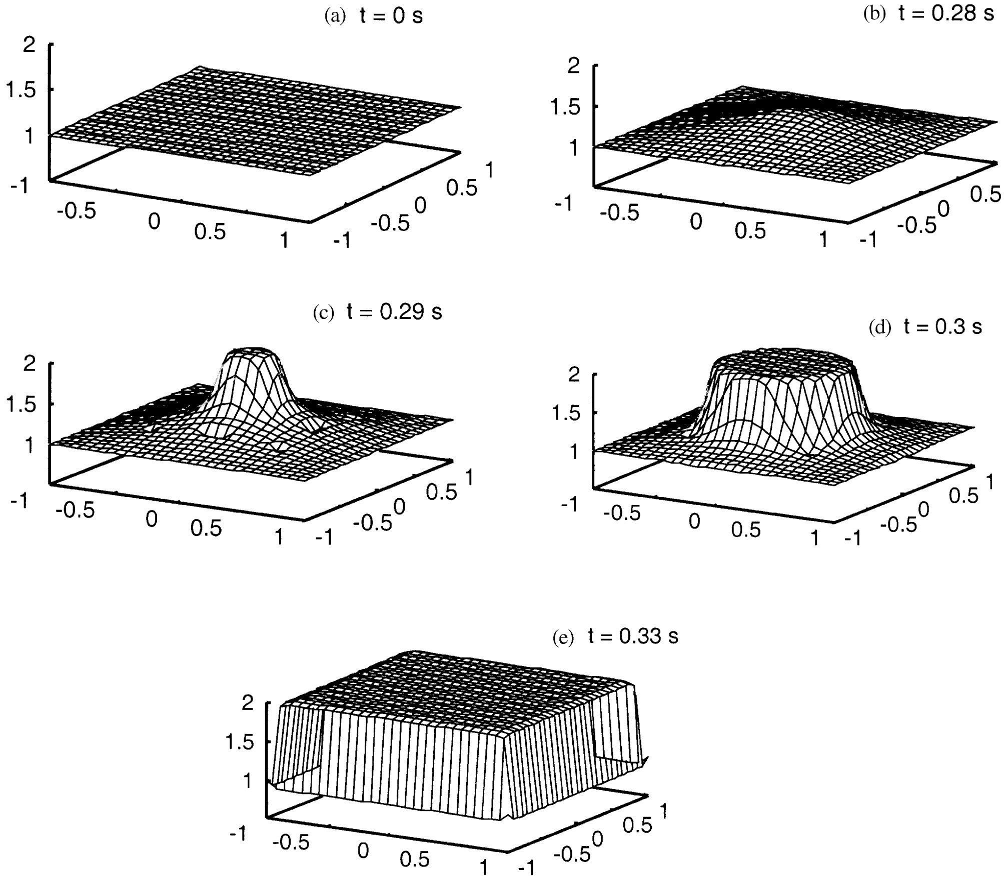


Fig. 10. Temperature vs (x,y) at several time steps for two-dimensional reaction model.

where $W(x, y; t)$ is vertical displacement, $D$ is stiffness and $m$ is mass per unit area. If the plate is simply supported on the four sides, the boundary conditions are

$$\frac{\partial^2 W}{\partial x^2} = W = 0 \quad \text{at} \quad x = 0 \text{ and } x = 1, \tag{19b}$$

$$\frac{\partial^2 W}{\partial y^2} = W = 0 \quad \text{at} \quad y = 0 \text{ and } y = 1. \tag{19c}$$

This problem can be analytically solved using Laplace transform. The analytical solution is, in the form of convolution integral, given by

$$W(x, y; t) = A(t)\sin(2\pi x)\sin(\pi y), \tag{20a}$$

$$A(t) = \int_0^t g(\tau)\cos\beta(t - \tau)\,\mathrm{d}\tau. \tag{20b}$$

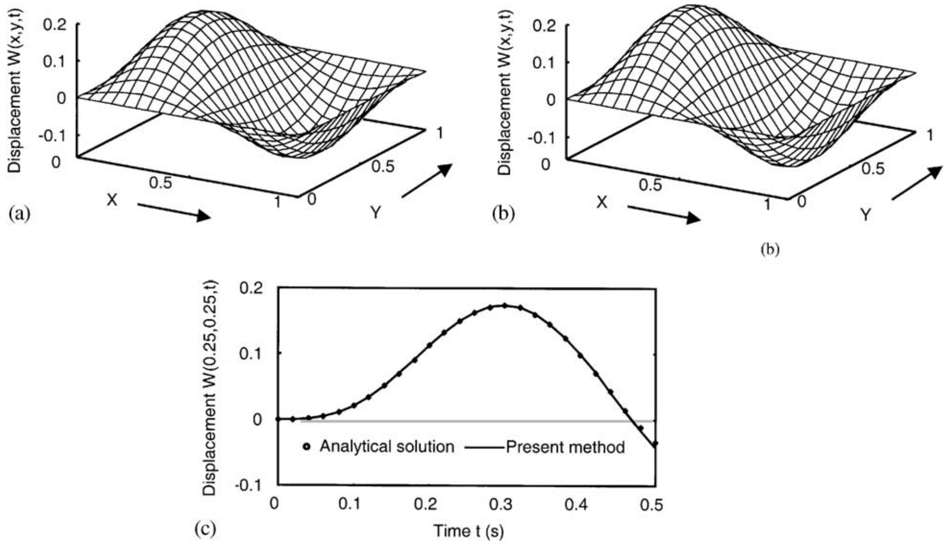


Fig. 11. Results obtained from (a) present method and (b) analytical solution at time step $t = 0.26$ s. (c) Comparison of time history of displacement obtained from present method and analytical solution.

Suppose $g(t)$ is a sinusoidal function of time, that is, $g(t) = G\,sin(\omega t)$. We obtain from Eq. (20b)

$$A(t) = -\frac{G\omega \sin \beta t}{\beta(\beta^2 - \omega^2)} + \frac{G \sin \omega t}{\beta^2 - \omega^2}. \tag{21}$$

Direct solution of Eq. (19) using current approach is shown in Fig. 11. The parameters used in the computation are: $D/m = 10^{-2}$, $G = 10$ and $\omega = 15$. The node number is $N_x = N_y = 20$, and $M = 7$, $M_p = 0$.

Fig. 11(a) shows the displacement field at time $t = 0.26$ s obtained from current approach. The analytical results at the same time step are shown in Fig. 11(b). A careful comparison shows that both are in very good agreement. Fig. 11(c) shows the comparison of time–history at the point (0.25, 0.25). The solid line denotes numerical result and the dots denote analytical solution. They are coincident.

## 5. Discussions

In this section, several potentially controversial issues are discussed.

There exists an optimum grid distribution when DQ is applied, which is often characterized by non-uniformity as given by Eq. (13). Instead of using a uniform grid distribution in the string vibration problem, see Eq. (7), will a non-uniform distribution like Eq. (13) give rise to dynamic numerical instability, too? The answer is yes. Non-uniform distribution can increase accuracy, but cannot avoid or remove dynamic numerical instability completely if the grid number $N$ is large. DQ is equivalent to differentiation using Lagrange interpolation. In the theoretical framework of Lagrange interpolation, it can be shown that the divergence near the end of the interval will actually grow exponentially as the number of points is increased [17]. Thus, the easiest way is to use a small number of nodes to approximate the derivatives at cortical nodes to prevent the exponential growth.

Classification of core and cortical nodes is crucial in applying the variable order approach presented here. There is a lack, however, of a strict procedure to determine the values for $M$ and $M_p$. It is difficult to provide theoretical values for them but the rule of thumb given in the previous section (that is, $M = 6-15$ and $M_P = 0-5$ ) will suffice practical applications. $M_p$ plays the role of smooth transition from a cortical node to a core node. $M_p = 0$ means a hard transition while a larger $M_p$ means a soft transition. $M_p$ needs not be too large. So the suitable range for it is $M_p = 0-5$.

One word should be mentioned on how to implement boundary conditions. In the examples treated in this paper, the boundary conditions are of same type, that is, the function under consideration takes on a fixed value on boundaries. This is easy to be implemented. At each time step in using Runge–Kutta scheme, the function values on boundaries are set to the given values. For general cases, however, more efforts are needed to implement boundary conditions. Many papers discussed this problem, say Ref. [2], which is thus neglected herein.

## 6. Conclusions

When DQ is applied, large errors may occur if the grid number is large. The errors exponentially grow as the grid number increases. Nodes near and on boundaries dominate the accuracy of DQ method. Dynamic numerical instability is caused by those nodes.

An easy way is introduced in this paper to remove the divergence near boundaries by applying a small number of points for cortical nodes. The method yields very good results for both linear and strongly non-linear dynamic problems. It significantly improves the accuracy of conventional DQ method.

## References

[1] R.E. Bellman, B.G. Kashef, J. Casti, Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations, Journal of Computational Physics 10 (1972) 40–52.

[2] C. Shu, Differential Quadrature and its Applications in Engineering, Springer, Berlin, 2000.

[3] S. Moradi, F. Taheri, Differential quadrature approach for delamination buckling analysis of composites with shear deformation, American Institute of Aeronautics and Astronautics Journal 36 (1998) 1869–1873.

[4] C.N. Chen, Differential quadrature finite difference method for structural mechanics problems, Communications in Numerical Methods in Engineering 17 (2001) 423–441.

[5] W.L. Chen, A.G. Striz, C.W. Bert, High-accuracy plane stress and plate elements in the quadrature element method, International Journal of Solids and Structures 37 (2000) 627–647.

[6] F. Civian, C.M. Sliepcevich, Differential quadrature for multidimensional problems, Journal of Mathematical Analysis and Applications 101 (1984) 423–443.

[7] J.R. Quan, C.T. Chang, New insights in solving distributed system equations by the quadrature methods—I, Computers and Chemical Engineering 13 (1989) 779–788.

[8] J.R. Quan, C.T. Chang, New insights in solving distributed system equations by the quadrature methods—II, Computers and Chemical Engineering 13 (1989) 1017–1024.

[9] C.W. Bert, M. Malik, The differential quadrature method for irregular domains and application to plate vibration, International Journal of Mechanical Sciences 38 (1996) 589–606.

[10] C.W. Bert, M. Malik, Differential quadrature method in computational mechanics: a review, Applied Mechanics Reviews 49 (1996) 1–28.

[11] H. Zengh, C.W. Bert, Adifferential quadrature analysis of vibration for rectangular stiffened plates, Journal of Sound and Vibration 241 (2001) 247–252.

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

[13] C. Shu, W. Kha, Numerical simulation of natural convection in a square cavity by SIMPLE-generalized differential quadrature method, Computers and Fluids 31 (2002) 209–226.

[14] C. Shu, W. Chen, H. Xue, H. Du, Numerical study of grid distribution effect on accuracy of DQ analysis of beams and plates by error estimation of derivative approximation, International Journal in Numerical Methods in Engineering 51 (2001) 159–179.

[15] G.D. Smith, Numerical Solution of Partial Differential Equations: Finite Difference Methods, Clarendon Press, Oxford, 1985.

[16] S. Adjerid, J.E. Flaherty, Amoving finite element method with error estimation and refinement for one-dimensional time dependent partial differential equations, SIAM Journal of Numerical Analysis 23 (1986) 778–795.

[17] K.E. Atkinson, An Introduction to Numerical Analysis, 2nd Edition, Wiley, New York, 1989.

[18] M.HC arpenter, D. Gottlieb, S. Abarbanel, Time-stable boundary-conditions for finite-difference schemes solving hyperbolic systems—methodology and application to high-order compact schemes, Journal of Computational Physics 111 (2) (1994) 220–236.