This equation indicates that the weighting coefficients for the high order derivative can be computed by the matrix multiplication of the weighting coefficients of the first order derivative. However, this equation is simple and involves more arithmetic operations as compared to equations (78 and 79). We noted that the calculation of weighting coefficient by equation (80) involves N multiplications and ($N-1$) additions, i.e., a total of ($2N-1$) arithmetic operations. Recurrence relationship (78) only involves two multiplications, one division, and one subtraction, i.e. a total of four arithmetic operations for calculation of each off-diagonal weighting coefficient, which is independent of the number of grid points N. The calculation of each diagonal weighting coefficient from equation (79) involves ($N-1$) subtractions. Thus, the number of arithmetic operations for equation (78) and equation (79) is substantially smaller than what is in equation (80).

Sample of typical grid distributions

Because the described equations obtained by using differential quadrature method are equivalent to one obtained by using quasi-spectrum method, the choice of grid points have a great effect upon accuracy of results. There are two kinds of methods for choosing the mesh points.

The uniform grid points are used in the first kind as follows:

Type (I): By a uniform grid, we mean that the grid has the same sizes. Thus by

setting $\Delta x = x_2 - x_1 = x_i - x_{i-1} = x_N - x_{N-1}, ...ect.$

The coordinates of the grid points are chosen as

$$x_i = (b-a)\frac{i-1}{N-1} \qquad \text{for } i = 1,2,....., N \text{ and } x_i \in [a,b].$$

The zeros of orthogonal polynomials such as Chebyshev polynomials are taken as grid points in the second kind as follows:

Type (II) : For this kind, the coordinates of the grid points are chosen as

$$x_i = \frac{b-a}{2}\frac{r_i - r_1}{r_N - r_1}, \quad \text{such that} \qquad r_i = \cos(\frac{i-1}{N-1}\pi)$$

In this field, there are some contribution studies about the effect of grid spacing distribution on the numerical results that were obtained by DQ method. Quan and Chang (1989) compared numerically the performances of the often-used non-uniform meshes and concluded that the grid points originated from the Chebyshev polynomials of the first kind is optimum in all cases examined. Bert and Malik (1996) indicated an important fact that the preferred type of grid points changes with problems of interest and recommended the use of Chebyshev-Gauss-Lobatto grid for structural mechanics computations. Maradi and Taheri (1998) also investigated the effect of various spacing schemes on the accuracy of DQ results for buckling application of composites. They provided insights into the influence of a number of sampling points in conjunctions with various spacing schemes. Chen (1997) and Bert and malik (1996) have provided sensible explanations why a certain type of grid points is superior to the others in the computation of their problems. The details of properties of DQ weighting coefficient matrices for the determination and rank are given by Shu (2000), and we note from this reference that these properties can be derived from the matrices properties in algebraic subject.

Exercise: if weighting coefficients are desired for a range $0 \le x \le 1$, then calculate the weighting coefficients matrices $C_{ik}^{(1)}$ and $C_{ik}^{(2)}$ for $N = 3,4,5$ grid points divided the above range.

Numerical methods to solve DQ resultant equations

It is very important to make simple review about the solution techniques, which are used to update the DQ resultant for the differential equations. In most applications of the DQ method to engineering and physics problems, which are governed by the partial differential equations, considering the second –order partial differential equation as follows:

$$\frac{\partial u}{\partial t} = f(t, x, u, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}) \text{.......................................} (81)$$

In general, Equation (81) should be specified with proper initial and boundary conditions for the solution to a specific problem. By DQ method at all interior points of whole domain, the original problem, which is defined in equation (81) can be reduced

to a set of N ordinary differential equations(ODEs) as

$$\frac{du(t,x_i)}{dt} \cong f(t, x_i, u(t,x_i), \sum_{k=1}^{N} C_{ik}^{(1)} u(t,x_k), \sum_{k=1}^{N} C_{ik}^{(2)} u(t,x_k)) \text{ for } i = 1,2,\ldots\ldots, N \text{ .}(82)$$

When $\frac{\partial u(t,x_i)}{\partial t} = 0$, we can be able to obtain a system of linear algebraic equations. The solution of partial differential equations may not be possible to express in closed-form. Therefore, this solution function can be approximated by polynomial approximation. Rearranging equation (82) to obtain a set of ordinary differential equations as;

$$\frac{d\{u\}}{dt} + L_{dq}\{u\} = \{G\} \text{ ....} (83)$$

where $\{u\}$is a vector representing a set of unknown functional values at all interior points, $L_{dq}\{u\}$ is a vector resulting from DQ discretization, $\{G\}$ is a vector arising from the given initial and boundary conditions. For time-dependent problems, equation (83) constitutes standard form ordinary differential equations. The time derivative can be approximated by explicit or implicit low order finite difference scheme. From equation (83), we can obtain a system of algebraic equations in the form

$$[H]\cdot\{u\}=\{G\} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (84)$$

where $\{u\}$is a vector of unknown functional values at all the interior grid points given by

$$\{u\}=\left(u_{2,2},u_{2,3},\ldots\ldots,u_{2,M-1},u_{3,2},u_{3,3},\ldots\ldots,u_{3,M-1},\ldots\ldots\ldots,u_{N-1,2},u_{N-1,3},\ldots\ldots,u_{N-1,M-1},\right)^{T_r}$$

and $\{G\}$ is a known vector given by

$$\{G\}=\left(G_{2,2},G_{2,3},\ldots\ldots,G_{2,M-1},G_{3,2},G_{3,3},\ldots\ldots,G_{3,M-1},\ldots\ldots\ldots,G_{N-1,2},G_{N-1,3},\ldots\ldots,G_{N-1,M-1},\right)^{T_r}$$

The dimension of the matrix $[H]$ is $(N-2)(M-2)$ by $(N-2)(M-2)$. Equation (84) can be written alternatively as

$$[C]\,[u]+[u]\,[D]=[G] \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (85)$$

this equation is called Lyapunov matrix form and $[C]$,$[D]$are matrices of weighting coefficients for the first and the second-order derivatives have the dimension $(N-2)(N-2),(M-2)(M-2)$ respectively. One can see that the dimensions of $[C]$ and $[D]$are very small compared with the dimensions of$[H]$. To solve this system that is discritized by DQ method, one can adopt direct method or iterative method. To solve the ordinary differential equations that are given in equation (83 or 84), there are different explicit numerical schemes that are used to discritize these equations and compute the results, for example: Euler forward explicit scheme; this is the first order scheme given by

$$u^{n+1}=u^{n}+\Delta t\cdot f^{n} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (86)$$

The solution techniques that we thought could be possibly used to solve the algebraic equations that are because of employing DQ method in governing equations are divided into two parts. The first part is named direct methods, and the second one is iterative methods.

Direct methods

To solve algebraic equations included in equation (84), there are many standard methods, amongst of them, Gaussian elimination method, LU decomposition approach are used extensively. The details of these methods can be found in textbook of numerical analysis. These methods are very efficient when the dimension of the matrix is not large. However, when the number of grid points increases the dimension of the matrix will increase accordingly. Hence the problem of virtual storage will become critical;

furthermore, the DQ discretization matrix tends to become ill-conditioned when the mesh size is large. This would lead to difficulties in obtaining the solution or even worse, reduce the accuracy of the solutions. The drawbacks of direct methods can be eliminated by using iterative methods. Some of these iterative methods have been used to solve the system of algebraic equations given in form of equation (84).

Iterative methods

If the matrix $[H]$ in equation (84) is composed of two matrices $[A]$ and $[P]$, then we can write it as

$$[H]=[A]+[P] \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(87)}$$

by rewriting equation (84) in terms of matrices $[A]$ and $[P]$, we obtain

$$[A]\cdot\{u\}=\{G\}-[P]\cdot\{u\} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(88)}$$

The iterative expression for equation (88) can be written as

$$[A]\cdot\left(\{u^{n+1}\}-\{u^{n}\}\right)=\{G\}-[P]\cdot\{u^{n}\}-[A]\cdot\{u^{n}\} \ \ldots\ldots\ldots\text{(89)}$$

where n represents iterative level and the right side sometimes is called the vector of the residuals. In practical applications, a relaxation factor Θ is introduced on the right hand side of equation (89), and the final iteration expression becomes

$$\{u^{n+1}\} = \{u^n\} + \Theta \cdot [A]^{-1} \cdot \{R^n\} \quad \text{..............................} \ (90)$$

such that

$$R^n = \{G\} - [H] \cdot \{u^n\}$$

Equation (90) is a general iterative expression for equation (84). By using different forms of $[A]$, we can obtain different iterative expressions for equation (84). For the
stability and convergence of iterative method the reader may consult the textbooks Smith (1978) and Rao (2002).

<u>Successive over-relaxation (SOR) iteration method</u>: SOR iteration is used to improve the convergence speed of Jacobi method. It is noted that SOR is a point iteration method. The value of u_i^{n+1} can be evaluated, when the values of $u_k^{n+1}, k = 1,2,.....,i-1$ is calculated. These new values at the iteration level $(n+1)$ can then be used to compute the residuals. The residuals of SOR iteration are computed from

$$\{R^n\} = \{G\} - [H_L]\{u^{n+1}\} - ([H_D] + [H_U])\{u^n\}$$

where $[H_L]$ is the lower triangular matrix with diagonal elements being zero, $[H_U]$ is the upper triangular matrix with diagonal elements being zero, and $[H_D]$ is the diagonal matrix with elements being the diagonal elements of $[H]$. It is noted that the elements of

$[H_L],[H_U]$, and $[H_D]$ are equal to those of $[H]$ at the corresponding positions, that is

$$[H]=[H_L]+[H_U]+[H_D]$$

The iterative expression of SOR method is the same as the following equation

$$u^{n+1}=u^n+\Theta\cdot\frac{R_i^n}{a_{ij}}$$

the R_i^n in the SOR method can be expressed as

$$R_i^n=G_i-\sum_{j=1}^{i-1}a_{ij}u_j^{n+1}-\sum_{j=1}^{M}a_{ij}u_j^n$$

SOR iterative methods for the Lyapunov system (2.20) to update the solution can be write as

$$u^{n+1}=u^n+\frac{\Theta}{C_{ii}+D_{jj}}\Big(G_{ij}-\sum_{k=1}^{i-1}C_{ik}u_{kj}^{n+1}-\sum_{k=i}^{N}C_{ik}u_{kj}^n-$$
$$\sum_{k=1}^{j-1}D_{jk}u_{ikj}^{n+1}-\sum_{k=j}^{M}D_{jk}u_{ik}^n\Big)\;\ldots\ldots\ldots\ldots\ldots\ldots\ldots(91)$$

with the residuals relation in the form

$$[R^n]=[G]-[C_L][u^{n+1}]-([C_D]+[C_U])[u^n]-[u^{n+1}][D_L]-[u^n]([D_D]+[D_U])$$

where the matrices C's and D's are having the same defined matrices of H's, which are mentioned above.

Gauss-seidel iteration method: it is special case of SOR (successive over-relaxation) iteration when Θ is taken as 1. There are many iterative methods some are related with these methods and others are different like Jacobin method, Jacobin over relaxation iteration method, Richardson iteration method, Conjugate Gradient iteration method…etc.

Error Analysis of DQ Method:

Shu (1991) and Chen(1996) are introduce the error resulting from approximation a function and its derivatives.

Error analysis for the function:

When approximate $u(x)$ by a polynomial of degree $(N-1)$, particularly by the Lagrange interpolation polynomial

$$\mathrm{P}_n u = \sum_{i=1}^{N} u(x_i)\cdot r_i(x) \qquad \ldots\ldots\ldots\ldots\ldots\ldots(92)$$

where $r_i(x)$ is the Lagrange interpolation polynomial given by

$$r_k(x) = x^{k-1} \qquad ,k = 1,2,\cdots\cdots\cdots,N$$

$$r_k(x) = \frac{\ell_N(x)}{(x-x_k)\,\ell_N^{(1)}(x)} \qquad ,k = 1,2,\cdots\cdots\cdots,N$$

$$\ldots\ldots\ldots(93)$$

$$r_k(x) = \frac{M_N(x)}{(x-x_k)\,M_N^{(1)}(x)} \qquad ,k = 1,2,\cdots\cdots\cdots,N$$

$$r_k(x) = 1,\ r_k(x) = (x-x_{k-1})\cdot r_{k-1}(x), \qquad ,k = 1,2,\cdots\cdots\cdots,N$$

Where ℓ is the Legender polynomial of degree N, and $M(x)$ defined as,

$$M(x) = (x-x_1)(x-x_2)\cdots\cdots\cdots(x-x_N) \qquad \ldots\ldots\ldots(94)$$

Thus, $M^{(1)}(x) = \prod_{k=1,k\neq i}^{N}(x_i - x_k)$

The approximate error of $u(x)$ is defined as,

$$\mathrm{E}(u) = u(x) - \mathrm{P}_N u \qquad \ldots\ldots\ldots\ldots(95)$$

If the Nth order derivative of the function $u(x)$ is assumed to be a constant, say k, then $u(x)$ can be expressed as

$$u(x) = m_0 + m_1 x + m_2 x + \cdots\cdots\cdots\cdots + m_{N-1}x^{N-1} + \frac{k\cdot x^N}{N!} \qquad \ldots\ldots\ldots\ldots(96)$$

Since equation(92) is exactly satisfied for a polynomial of degree less than or equal $(N-1)$, we have

$$\mathrm{E}(x^k) = 0, \qquad k = 0,1,\cdots\cdots\cdots\cdots,N-1 \qquad \ldots\ldots\ldots\ldots(97)$$

Substituting equation(96) into equation(95) and using equation(97), we obtain

$$\mathrm{E}(u) = k\cdot\frac{\mathrm{E}(x^N)}{N!} \qquad \ldots\ldots\ldots\ldots\ldots(98)$$

$$where \quad \mathrm{E}(x^N) = x^N - \sum_{i=1}^{N} x_i^N \cdot r_i(x)$$

On the other hand, substituting the polynomial of degree $(N-1)$, $g(x) = x^N - M(x)$ into equation(95),we obtain

$$\mathrm{E}(g) = x^N - M(x) - \sum\left[x_i^N - M(x_i)\right]\cdot r_i(x) = 0 \qquad \ldots\ldots\ldots\ldots\ldots(99)$$

Since, $M(x_i) = 0$, equation(99) can be further reduced to

$$x^N - \sum x_i^N \cdot r_i(x) = M(x_i) \qquad \ldots\ldots\ldots\ldots\ldots(100)$$

Finally, substituting equation (100) into equation (98), we get

$$\mathrm{E}(u) = \frac{k \cdot M(x_i)}{N!} \qquad \ldots\ldots\ldots\ldots\ldots(101)$$

In most cases, the Nth-order derivative of the function $u(x)$ is not a constant, but it may be boundad. In this case, we can adopt another method to analyze $\mathrm{E}(u)$. For simplicity, we set $\phi(x) = \mathrm{P}_N u$ and defined a function $U(z)$ as,

$$U(z) = u(z) - \phi(z) - C \cdot M(z) \qquad \ldots\ldots\ldots\ldots\ldots\ldots(102)$$

Clearly, when $z = x_1, x_2, \cdots\cdots, x_N$, $U(z) = 0$.

If we set $U(x) = 0$,we obtain

$$\mathrm{E}(u) = u(x) - \mathrm{P}_N u = u(x) - \phi(z) = C \cdot M(x) \qquad \ldots\ldots\ldots\ldots\ldots(103)$$

Since $U(z)$ has $N+1$ roots $x_1, x_2, \cdots\cdots, x_N$ in the domain, by repeated application of Roll's theorem, the Nth-order derivative of $U(z)$, $U^{(N)}(z)$ is found to have at least one root lying between x_1 *and* x_N. Denoting this root by ξ,we have

$$U^{(N)}(\xi) = 0 \qquad \ldots\ldots\ldots\ldots\ldots(104)$$

Not that, $\phi(z)$ is a polynomial of degree $(N-1)$.so from equation(102), we obtain

$$C = \frac{u^{(N)}(\xi)}{N!} \qquad \ldots\ldots\ldots\ldots\ldots\ldots(105)$$

Hence

$$\mathrm{E}(u) = \frac{u^{(N)}(\xi) \cdot M(x)}{N!} \qquad \ldots\ldots\ldots\ldots\ldots\ldots(106)$$

In general ξ is a function of x.

Error analysis for the derivatives:

The error for mth-order derivative approximation can be defined as

$$\mathrm{E}_D^{(m)}(u) = \frac{\partial^m u}{\partial o^m} - \frac{\partial^m(\mathrm{P}_N u)}{\partial o^m} = \frac{\partial^m u}{\partial o^m} - \frac{\partial^m \phi}{\partial o^m} \qquad \ldots\ldots\ldots\ldots\ldots\ldots(107)$$

Where $m = 1,2,\cdots\cdots, N-1$. Using equation (106), equation(107) can be written as

$$\mathrm{E}_D^{(m)}(u) = \frac{\partial^m \left[u^{(N)}(\xi) \cdot M(x) \right]}{N! \ \partial x^m} \qquad \ldots\ldots\ldots\ldots\ldots\ldots(108)$$

Since ξ is an unknown function of x, it is difficult to estimate $\mathrm{E}_D^{(m)}(u)$ using equation(108).as a special case, if we assume that the Nth-order derivative of $u(x)$ is a constant, say k,equation(108) can be simplified to

$$\mathrm{E}_D^m(u(x_i)) = \frac{k \cdot M^{(m)}(x_i)}{N!} \quad \ldots\ldots\ldots\ldots\ldots\ldots(109)$$

For the general case where $u^{(N)}(\xi)$ is not a constant, we can use a similar method as in the analysis of the function approximation to conduct error analysis of the derivative approximation.

Since $g(z) = u(z) - \phi(z)$ has N roots in the domain, according to Roll's theorem, its mth-order derivative $g^{(m)}(z)$ has at least $(N-m)$ roots in the domain, namely $\bar{x}_1, \bar{x}_2, \cdots\cdots, \bar{x}_N$. Thus, the function

$$\begin{aligned} U^{(m)}(z) &= g^{(m)}(z) - \bar{C} \cdot \bar{M}(z) \\ &= u^{(m)}(z) - \phi^{(m)}(z) - \bar{C} \cdot \bar{M}(z) \end{aligned} \quad \ldots\ldots\ldots\ldots(110)$$

where $\bar{M}(z) = (z - \bar{x}_1)(z - \bar{x}_2)\cdots\cdots(z - \bar{x}_{N-m})$,would vanish $\bar{x}_1, \bar{x}_2, \cdots\cdots, \bar{x}_{N-m}$. Now if we set $U^{(m)}(\bar{x}) = 0$,where $\bar{x}$ is different from $\bar{x}_1, \bar{x}_2, \cdots\cdots, \bar{x}_{N-m}$, then $U^{(m)}(z) = 0$ has $(N-m+1)$ roots ,and

$$\mathrm{E}_D^{(m)}[u(\bar{x})] = u^{(m)}(\bar{x}) - \phi^{(m)}(\bar{x}) = \bar{C} \cdot \bar{M}(\bar{x}) \quad \ldots\ldots\ldots\ldots\ldots\ldots(111)$$

Using Roll's theorem repeatedly the $(N-m)th$-order derivative of $U^{(m)}(z)$ is found to have at least one root $\bar{\xi}$. Thus equation(111) can be reduced to

$$\mathrm{E}_D^{(m)}[u(\bar{x})] = \frac{u^{(N)}(\bar{\xi}) \cdot \bar{M}(\bar{x})}{(N-m)!} \quad \ldots\ldots\ldots\ldots\ldots\ldots(112)$$

Equation(112) can be used to estimate the error of the derivative approximation. It is assumed that all the coordinates are in the interval Δx,and the Nth-order derivative of the function $u(x)$ is bounded, then

$\left|u^{(N)}(\bar{\xi})\right| \le C$, where C is a positive constant, and

$$\left|M^{(m)}(x)\right| \le N(N-1)\cdots\cdots(N-m+1)(\Delta x)^{N-m}$$

$$\left|\bar{M}(\bar{x})\right| \le (\Delta x)^{N-m}$$

So, equation(109) and equation(112) can be simplified to

$$\mathrm{E}_D^{(m)}[u(\bar{x})] = \frac{C \cdot (\Delta x)^{N-m}}{(N-m)!}, \quad for\ m = 1, 2, \cdots\cdots, N-1 \quad \ldots\ldots\ldots\ldots\ldots(113)$$

For the general case, the error distribution of the derivative approximation can also be studied using equation (108). The error distributions of the first-, second- , third- and fourth- order approximation have studied by Chen (1996). For the first-order derivative, equation (108) gives

$$\mathrm{E}_D^{(1)}(u) = \frac{u^{(N+1)}(\xi) \cdot \xi_x \cdot M(x) + u^{(N)}(\xi) \cdot M^{(1)}(x)}{N!} \quad \ldots\ldots\ldots\ldots\ldots\ldots(114)$$

Let $k_1 = \max\left\{\left|u^{(N)}(\xi)\right|\right\}$ and note that $M(x_i) = 0$. Applying equation (114) at that grid point x_i gives

$$\mathrm{E}_D^{(1)}\left[u(x_i)\right]=\frac{u^{(N)}(\xi)\cdot M^{(1)}(x_i)}{N!}\leq k_1 e^{(1)}(x_i) \quad , for \;\; i=1,2,\cdots\cdots,N \qquad \ldots\ldots(115)$$

where $e^{(1)}(x_i)=\frac{M^{(1)}(x_i)}{N!}$, is the error distribution of the first order derivative approximation. For the second-order derivative

$$\mathrm{E}_D^{(2)}(u)=\frac{2u^{(N+1)}(\xi)\cdot\xi_x\cdot M(x)+u^{(N)}(\xi)\cdot M^{(2)}(x)}{N!}\leq k_2 e^{(2)}(x) \qquad \ldots\ldots\ldots(117)$$

Where $k_2 = \max\left\{\left|u^{(N)}(\xi)\right|, \left|\xi_x u^{(N+1)}(\xi)\right|\right\}$, and $e^{(2)}(x_i)$, is the error distribution of the second order derivative approximation.