Dissertation for the Doctoral Degree of Philosophy

Differential Quadrature Method and its Applications in Engineering

– Applying Special Matrix Product to Nonlinear Computations and Analysis

Wen Chen

Advisor: Tingxiu Zhong

Specialty: Mechatronic Control and Automation
Department of Mechanical Engineering
Shanghai Jiao Tong University

December 22, 1996

Dedication

To my parents and their love.

致　谢

本篇论文是在钟廷修教授的悉心指导下完成的。三年来钟廷修教授不倦的教诲和生活上的支持，使作者不仅在学业上有很大的进步，也领悟了不少有益的生活哲理。导师的教导对作者的影响无疑是深远的。南京航空航天大学的王鑫伟教授，本校力学系张永元教授，机械工程系的张建武教授和数学系的木定夷教授在论文方面给予了许多指导和有力的帮助。在此深表感谢。此外，美国Oklahoma大学的C. W. Bert教授及时地提供给作者他在这个领域的一些有意义的论文，在此也表示感谢。

作者也感谢钱则明，郑志令老师以及93博B班全体同学在作者学习期间的友好帮助和有益的讨论。

最后感谢父母多年来的养育之恩和无私的奉献，使作者能够顺利地完成学业¡

ABSTRACT

Numerical solution procedures occupy an extremely important part in many areas of science and engineering. The subjects in these fields are of continued research interest. A variety of numerical methods are available today for solving the initial- and/or boundary value problems in physical science and engineering. In recent years, the differential quadrature (DQ) method has become an increasingly popular numerical technique for the rapid and efficient solution of a variety of science and engineering problems. Some shortcomings in the conventional numerical techniques are not inherent in the method. Due to its rather recent origin, the DQ method may be not well known to the engineering community. In this paper, we study the DQ method and its applications, and introduce two kinds of special matrix product into nonlinear computations and analysis of the DQ method as well as other numerical methods. Our work is a step forward in nearly all important basic aspects of the method and its applications. In the following we will briefly state our main contributions.

First, the conventional formulas of the truncation error in the DQ method do not involve the practical grid interval and are too imprecise for many practical applications. We have obtained new truncation error formulas, which can more accurately estimate the truncation error at any grid point than before. Through the numerical trial and error, we propose seven general rules for choosing grid spacings. Since the DQ method is in fact equivalent to the collocation and pseudo-spectral methods, the truncation error formulas and rules for choice of sampling points are also applicable for these two methods. We also give the simplified formulas for computing the DQ weighting coefficients under equally spaced grid points and the zeros of the Chebyshev polynomials or Legendre polynomials, and overcome the difficulty that the conventional applications of zeros of orthogonal polynomials can not encompass the boundary points. Second, we give the DQ approximate formulas in matrix form for boundary value problems instead of traditional polynomial ones. By using these formulas, the fast algorithms in the solution of Lyapunov matrix equation are introduced to the DQ calculations of multi-dimensional problems of the second order with three orders of magnitude less computing time. The efficiency and simplicity of the presented technique were demonstrated in the DQ solution of the Poisson equation and steady convection-diffusion equations. We also notice the fact that the rank of the DQ weighting coefficient matrices for the ith order derivative is M-i, where M is the number of grid points. Moreover, the coefficient matrix is in fact power zero matrix. Based on this fact, we proposed a new approach in implementing the multiple boundary conditions in the DQ solution of boundary value problems of more than two order, which are often encountered in structural engineering analysis. We also solve the difficulty applying the boundary conditions at corner points for this approach. Compared with the other existing approaches, the presented approach shows easy use, good stability, wide applicability and high accuracy.

Third, we first applied the DQ method to the initial value problems and pointed out that the method is A-stable. Therefore, the DQ method is always stable for stiff problems. Also, the DQ method is unconditionally stable for structural dynamic analysis. We presented the DQ approximate formulas in matrix form for initial value problems. Thus, the formulations of initial value problems can be expressed as Lyapunov algebraic matrix equation. Several fast algorithms in the solution of the Lyapunov matrix equations are successfully applied to reduce the computing effort and storage requirements by an order of N^3 and N^2, respectively, where N is the number of interior grid points. Consequently, the DQ method requires nearly the same computational effort in the solution of linear stiff problems as the existing single step methods such as Newmark and Gear methods, etc. while its high computational efficiency is maintained. Fourth, we proved that the weighting coefficient matrices of the DQ and HDQ methods are centrosymmetric or skew centrosymmetric matrices if using symmetric grid spacing. The centrosymmetric matrices can be factorized into tow smaller size matrices in the evaluation of its inverse, determinant, eigenvectors and eigenvalues. Therefore, the computational effort and storage requirements in applying the DQ method for certain problems can be reduced by 75 per cent and 50 per cent, respectively.

Finally, it may be our most important contributions that the Hadamard product and SJT[1] product are introduced to the nonlinear computations of the DQ method as well as other numerical methods, including finite element, finite difference, boundary element, spectral, pseudo-spectral, Runge-Kutta, Newmark, Wilson θ and Gear methods, etc. Due to the application of the concept of Hadamard product and SJT product, the nonlinear formulation efforts are greatly reduced and programming task is simplified, and we obtain the formulation-H and -S as two kinds of the unified formulation form for general nonlinear computation. The simple iteration method becomes a very competitive technique to the Newton-Raphson method in the solution of general nonlinear algebraic equations. We also derived explicit formulas to describe the relation between the condition number and perturbed error bound for the nonlinear formulation. The SJT product is first defined by us to provide a very simple and highly efficient approach to compute the Jacobian derivative matrix in the Newton-Raphson method for the solution of the nonlinear formulations. Due to the utility of the Jacobian matrix in a wide range of science and engineering areas, the SJT product may have high potential for many theory and applied analysis. Also, the coupling formulations for the corresponding coupling nonlinear partial differential equations can be easily decoupled by means of the Hadmard product and SJT product. For example, the

[1] The SJT is abbreviate of Shanghai Jiao Tong University.

computational effort and storage requirements in the DQ analysis of geometrically nonlinear bending of orthotropic plates are reduced to about only one twenty-seventh and one-ninth, respectively, as much as those by Bert et al. due to the decoupling computations.

Matrix computations is of extreme importance to nonlinear analysis and computations. However, it is noted that the conventional linear algebra is based on the linear transformation. In fact, the ordinary matrix product computation seems not to undertake the task of nonlinear analysis and computations very well. We need seek an alternate route to handle nonlinear problems. The Hadamard product and SJT product provide nonlinear computations and analysis with a new framework. Compared with the standard matrix approach applied in the nonlinear computation and analysis, they greatly simplify the work. Due to the extreme importance of nonlinear computation and analysis, the present researches have substantial significance in theoretical and applied areas.

We also give a new approximate formula for directly computing the inverse of the Jacobian matrix in the Newton-Raphson method. The formulas involves only the ordinary matrix multiplications and, thus, can avoid the affect of the possible ill-conditioning of the Jacobian matrix on the convergence of the Newton-Raphson method. We also give a simple approach to eliminating or reducing the cross nonlinear algebraic terms in the resulting DQ formulations for some differential operators and, thus, the computational efficiency of the DQ and DC methods can be improved significantly. We discuss some essential problems for further developing the DQ method and Hadamard product and SJT product approaches into engineering numerical techniques.

In the appendices, we discuss algebraic and analysis properties of the Hadamard product. It is also proved that the weighting coefficient matrices of the quadrature method based on Fourier-type trigonometric functions are circulant matrices.

KEY WORDS: numerical method, differential quadrature method, Hadamard product, SJT product, Jacobian matrix, centrosymmetric matrix, Lyapunov matrix equation, truncation error, differential cubature method, stiff differential equations, simple iteration method, grid spacing.

博士论文扩展摘要

微分求积法及其工程应用
–特殊矩阵乘积在一般的非线性计算中的应用

陈文

指导教授：钟廷修

数值计算在广泛的科学和工程领域中很重要。虽然计算机硬件最近几年有了迅速的发展，但科学家和工程师对数值算法的发展抱有更高的期望。这是因为算法的改进和提高对计算和仿真的规模，精度和可靠性有更深刻的影响。微分求积法(Differential Quadrature Method, 英文简称DQ法)是最近几年引起广泛注意的一种数值计算方法。这个算法具有数学原理简单，使用方便，计算精度高，计算量和内存需求量少，不依赖泛函和变分原理，边界条件不必另外考虑，适合于微型机等优点，是瑞利－里兹法，迦辽金法，配点法和拟谱法的一种有竞争力的替代方法，有很好的发展前景。对规则域问题和分布式参数系统，这个方法比有限元，有限差分和边界元法有高得多的效率。最近，谱单元法中的多域法和坐标映射技术被用于DQ法计算不规域问题和钢架结构的分析，并取得了初步的成功。但这方面还有很多的工作要做。DQ法也被成功地用于计算有很大变化梯度的问题。此外，这个方法对非线性问题非常有效。在本文中我们研究了这个算法和它的一些工程应用，并将两种特殊矩阵乘积应用于DQ法和其他数值算法的非线性计算和分析。下面简述我们的主要工作。

第一，Bellman基于Rolle定理所给的DQ法截断误差公式和实际使用的节点位置无关，太不精确，不适合于实际应用。我们基于多项式插值公式导出了新的精确的DQ法截断误差公式。由我们的公式，可精确地计算DQ法在每个节点的截断误差，并发现DQ法是高阶收敛的算法(n-2阶，n是所用节点个数)。我们通过数值试验发现以切比雪夫多项式的根为节点并不是对所有问题都是最优的，虽然可能是目前已知节点分布中对很多问题最好的一种。从截断误差分析看，这是因为DQ法用该种节点分布在靠近端点的内节点的截断误差较在其他内点处大，等分点和勒让德点也有类似形式的截断误差分布。这不符合最优逼近所要求的截断误差的极大极小原则，即不同节点的截断误差的绝对值应该尽可能地一致。实际上，切比雪夫多项式是由函数的多项式最佳逼近导出的，并不是针对函数的导数逼近。通过一些数值试验，我们给出了七个选择DQ法节点分布的一般规则。应该强调的是问题的边界条件类型对DQ法用不同节点分布的稳定性和收敛速度有很大的影响。由于DQ法与配点法和拟谱法在本质上是相同的，因而，我们给出的DQ法截断误差公式和选择法节点分布的规则也适用于这两种算法。我们也给出了在标准变量域下简化的直接计算等分节点，切比雪夫和勒让德节点DQ法权系数的公式。由此公式，切比雪夫和勒让德节点自动包含了标准变量域的边界端点，克服了以往DQ法应用这两种节点分布时所遇到的一个主要困难。

第二，DQ法已被用于计算各种工程边值问题，并取得了很大成功。我们给出了DQ法解边值问题的矩阵形式的逼近公式以替代传统的多项式逼近公式，极大简化了Formulation工作量，并由此公式将求解Lyapunov代数矩阵方程的快速算法引进到DQ法解多维二阶偏微分方程中，使计算工作量和内存需要量大约减少了三个数量级。实际应用于Poisson方程和稳态热传导/扩散对流问题的计算也证实了我们的方法的简单和高效率。此外，我们发现DQ法的权系数矩阵为幂零矩阵，并据此提出了一种能够精确满足所有边界条件用于DQ法解4阶边界值问题(结构工程中经常遇到此类问题)的有效方法。我们

也克服了该方法解决多维问题时遇到的运用角点边界条件的困难。将我们的应用于结构元件的静态弯曲，振动和屈曲的计算分析显示，同已有的其他方法相比，该方法使用方便，适用面广，灵活，稳定性和精度较高。是一个非常有竞争力的方法。

第三，刚性问题在自动控制，电子 网 络，生物学，物理及化学动力学过程中经常遇到。传统的常微分方程的数值积分方法在解决这类问题时遇到极大困难，算法的计算稳定性是求解这类问题时主要考虑的因素，有关这方面的算法的研究目前非常活跃。另一方面，结构动力问题则在建筑，地质，海洋平台及机械设计与制造等许多领域有广泛的应用。这类问题一般也是刚性的。我们首次将DQ法用于初值问题(结构动力问题和刚性问题)的计算。我们指出DQ法是A稳定算法。因而，DQ法对刚性问题求解总是稳定的。这一点也被数值算例所证实。对于一般的结构动力分析问题，因为阻尼矩阵产生负的实特征根，所以DQ法也是无条件稳定的。我们对初值问题提出了DQ法的矩阵形式的逼近公式。因而，对线性问题最后得到的是Lyapunov代数矩阵方程。由此，我们将有关Lyapunov代数矩阵方程的快速求解算法引入到DQ算法初值问题的计算中，极大地减少了计算工作量(大约N^3倍，N为DQ法节点个数)，使得DQ法对线性问题的计算工作量大致等同于常用的其他算法的计算工作量。而DQ法是高阶收敛的，因而，DQ法在线性结构动力问题和刚性问题计算中远优于目 前常用的Newmark, Wilson θ, Houblt, Rung-Kutta和Gear等方法。对非线性问题则计算工作量要大于一般的单步法，但由于DQ法有高得多的精度和收敛速度，因而，计算成本并不高。这方面DQ法的计算效率可能还可以提高很多，如使用简单迭代法解非线性方程组等，还有大量的工作要做。DQ法计算边值问题，对复杂的不规则域问题还有一定的困难。但是对初值问题不存在这个问题，因而，这是DQ法很有意义的一个研究和应用方向。

第四，微分求积法和调和微分求积法(Harmonic Differential Quadrature Method, 英文简称HDQ法, 不同于微分求积法在于使用了三角函数为基函数)在实际应用中常用等分点和正交多项式（如切比雪 夫多项式和勒让德多项式)的根为节点。如此形式的节点为对称分布节点。我们证明了在此种情况下, DQ法和HDQ法权系数矩阵为中心对称矩阵。在求解中心对称矩阵的逆，特征值和特征向量时可将该矩阵分解成两个约为原矩阵一半规模的较小矩阵。因而，DQ和HDQ法对某些问题，特别是对带有对称边界条件的问题计算工作量可减少75%，内存需求量减少一半。对板的静态弯曲，振动分析和稳态热传导/扩散对流问题的计 算表明运用DQ法的权系数矩阵中心对称性质，可以减少计算工作量并扩大解题规模。

第五，最后，也是我们最重要的工作是将特殊矩阵乘积－Hadamard积引入到DQ法和HDQ法以及一般的非线性数值计算(包括有限元，有限差分，边界元，配点法，谱方法，拟谱法，Newmark, Wilson θ, Runge-Kutta, Gear法等)。利用Hadamard积，我们得到了两种清晰的矩阵形式的非线性数值逼近公式，这极大地减少了Formulation工作量，并方便了计算机编程。非线性数值计算的稳定性分析一直不是一个容易的任务。运用Hadamard矩阵积，我们导出了一般的非线性计算摄动误差的估计公式. 稳定性问题和摄动误差分析有密切的联系。因而，我们公式为进一步分析非线性计算的稳定性问题提供了基础。应用简单迭代法(Simple Iteration Method)计算非线性代数方程组有一个很大困难是构造高效率的迭代方程。标准的迭代格式效率往往不高。这是简单迭代法应用不如Newton-Raphson法广泛的一个重要原因。应用Hadamard矩阵积，我们能够象用简单迭代法求解一维标量非线性方程那样方便地对多维非线性方程组构造多种迭代方程，并确定最有效的迭代格式, 从而极大地改进了简单迭代法的使用效率使其成为与Newton-Raphson法相比有竞争力的算法。实际算例也证实了这一点。本

文的研究表明Hadamard积是描述非线性计算和分析问题强有力的工具。此外，大量有关Hadamard积的已有工作可以在非线性计算和分析中加以充分利用，无疑，进一步的研究工作是必须的。

我们也首次定义了一种新的特殊矩阵乘机－SJT[*]矩阵积. 运用SJT积, 我们获得了计算Newton-Raphson法中Jacobi导数矩阵精确解的简单，高效率的计算公式。鉴于Jacobi矩阵在许多问题中有重 要的应用, 因而, SJT积法有广泛的理论和应用价值. Hadamard积和SJT积法能够用于一般的耦合非线性微分方程的数值解耦计算, 即把相应的耦合数值逼近方程解耦, 这个解耦过程类似于耦合的标量方程的解耦计算, 非常简单容易。因而，计算工作量和内存需求量可以大大地减少。Hadamard积和SJT积法已被用于DQ法计算各向异性板的大桡度非线性弯曲问题和非线性结构动力和流体动力问题，并获得很大的成功，对各项异性板的非线性大桡度弯曲问题，由于解耦计算，计算工作量和内存需求量分别减少到 Bert等所用的二十七分之一和九分之一. 一个审稿人(关于投稿论文”The Study on Nonlinear Computations of the DQ and DC methods” by Chen and Zhong, in press in Inter. J. of “Numer. Methods for P. D. E.”, 1996)评价Hadamard积和SJT积技术，”which may revolutionize the practical implementation of the DQ and DC methods敗Ｎ颐侨衔猉adamard积和SJT积技术对其他的数值技术的非线性计算也将有重要的意义。

矩阵方法在许多领域（特别是在与计算机有关的领域〕发挥着非常重要的作用。目前常用的一般矩阵乘积是基于线性代数变换，因而从本质上讲不适合于非线性计算和分析。实际上，标准的线性代数和矩阵分析技术在非线性计算和分析中已显得力不从心。Hadamard积和SJT积是两种非常简单的特殊矩阵运算, 与目前非线性计算和分析中常用的其他 方法相比，极大地简化了所需的 工作。由于非线性计算和分析是广泛关心的课题，因而，这些工作有重要的理论和应用价值。

我们给出了直接计算Newton-Raphson法中Jacobi矩阵的逆矩阵的一个新的简单的近似公式。该公式仅需要做两次矩阵乘法运算， 不需要通常的Gauss消去法和Gauss-Jordan法求逆计算。因而，可以克服大的非线性系统迭代求解中常出现的病态Jacobi矩阵使Newton-Raphson法无法收敛的困难。通过将一个较复杂的非线性微分算子转化为一个线性算子和较简单的非线性算子的耦合算子方法，我们能够消除或减少该非线性算子的DQ逼近方程中的耦合代数项。从而极大地减少了计算工作量和内存需要量。最后，我们也讨论了将DQ法和Hadamard积与SJT积技术发展成一种实用算法有待解决的一些重要问题和相关的研究方向。在附录中，我们对Hadamard积的分析和代数性质做了广泛的讨论。我们也证明了以富利叶型的三角函数系为基函数的微分求积法的系数矩阵为循环矩阵。基于其他研究者和我们的最近的工作，我们相信DQ法在不久的将来能被发展成为一种实用的工程算法。

[*] SJT是Shanghai Jiao Tong University的英文缩写。

TABLE OF CONTENTS

CHAPTER 1.

INTRODUCTION

1.1. General Problem Statement

The numerical computations have been playing a prominent role in many science and engineering areas. The rapid advance in the design of computer has a profound affect on this research field, but scientists and engineers have more expectations for improvements in numerical methods than in computers. The development of computational methods may be of vital importance to increase the scale, accuracy and reliability of computations, since an inordinate amount of computing time and storage required often prohibits the calculation. On the other hand, the advances in microelectronics have also spurred remarkable expansion of engineering methodology and software. The lower cost of simulation has helped integrate numerical modeling more completely into the analysis and design sequence. An effective simulation capability is often the main factor in bringing a new engineering product to market in a timely manner or redesigning a system to meet desired objectives. Numerical simulation can also be an effective aid to physical experiments and theoretical analysis and is used to guide the development of new constitutive models and physical theories. In addition, the study of numerical computation itself can lead us to some of the deepest portions of pure and applied analysis. Therefore, the research in this field will be very significant and full of challenge.

A variety of numerical methods are available today for solving the initial- and/or boundary value problems in physical and engineering science, for example, finite difference (FD), finite element (FE), boundary element (BE), least square, Rayleigh-Ritz, Galerkin, pseudo-spectral, collocation, spectral, Newmark, Wilson θ, Houblot, Runge-Kutta, semi-implicit Runge-Kutta, implicit Runge-Kutta, and Gear methods, etc. These methods, however, have respective drawbacks in that they may be too complex mathematically for routine engineering analysis, be relatively easy to used but limited to special cases, need initial trial functions, or require large amounts of computational effort and consequently high cost. For instance, in a large number of problems, reasonably approximate solutions are desired at only a few specific points in the physical domain. However, in order to get results even at or around a point of interest with acceptable accuracy, traditional FE and FD methods still require the use of a large number of grid points. Consequently, the requirements for CPU time and storage are often unnecessarily large in such cases. On the other hand, the Rayleigh-Ritz and Galerkin methods require less computational effort as compared with the FE and FD methods. However, these methods require one to select initial trial functions satisfying boundary conditions for problems considered, while this is not an easy task in practice. In addition, these methods also need more strenuous formulation effort. Recently, the spectral and pseudo-spectral methods have been extensively used in engineering analysis especially for the numerical solutions of fluid dynamic problems. The spectral and pseudo-spectral methods belong to global numerical techniques and are efficient for many linear and nonlinear problems, but the desired spectral coefficients usually have no physical significance and thus assumed initial values in the solution of nonlinear problems are inherently poor guess, so the computational effort for the nonlinear analysis is aggravated. The BE method has definite intrinsic advantages for some kinds of engineering problems in comparison over other numerical techniques, but it is not well suited for highly nonlinear and inhomogeneous problems. For initial value problems, the Newmark, Wilson θ, and Houblot methods are unconditionally stable, but the accuracy of these methods are only of two order. The Runge-Kutta method is simple and efficient but only conditionally stable and, thus, limited to non-stiff systems. The Gear method is high order accurate and stiffly stable for many engineering computations. However, the method is not suitable for structural dynamic problems with very high frequency because it is not A-stable.

The above-mentioned shortcomings are not inherent in the differential quadrature (DQ) method. Due to its rather recent origin, the DQ method may be not well known to the computational mechanics community. The DQ method can yield highly accurate solutions with relatively little computational effort and storage requirements and, thus, are very suitable for recently popular personal computers. The method can easily and exactly satisfy a variety of boundary conditions and require much less formulation and programming effort. It have been pointed out that the DQ-type methods are basically equivalent to the collocation (pseudo-spectral) methods. But the DQ and DC methods directly compute function values at grid points rather than spectral variables. Thus, they are more explicit and simple for some practical applications and especially advantageous for nonlinear problems. Moreover, the mathematical techniques involved in the method are not sophisticated. So the DQ method is easily learned and used. The DQ method has been shown in many studies to be a very competitive alternative to the Rayleigh-Ritz, Galerkin, collocation and pseudo-spectral methods for general purpose. In general, the method is also much more efficient than the FE and FD methods for models of distributed-parameter systems and problems with regular geometries. The theoretical analysis and numerical experiments also show that the DQ method is especially efficient for highly nonlinear problems. Recently, the DQ method has be extended to handle irregular shaped problems as well as truss and frame structures, and achieved an early success. By employing the proper functions as basic functions, the method has been applied successfully to deal with problems involving steep gradients. Due to the above striking merits of the DQ method, in recent years the method has become increasingly popular in the numerical solution of problems in engineering and physical science.

However, some essential problems in the DQ method have been not involved or fully studied in the existing literature, for example, computational stability, truncation error, application for initial value problems and numerical properties of its weighting coefficient matrices. The applications of the method have so far been limited to smaller scale problems. The problems involved with nonlinearity have been seldom analyzed by the DQ methods yet.

On the other hand, the nonlinear numerical computations and analysis become more important in recent years. The subject in this field is another major purpose of this study. The traditional linear algebraic approach, which is very successful for linear numerical computations, has been extended to handle the nonlinear problems. However, since nonlinear problems have actually different from linear ones, linear algebraic and the relative matrix approaches, which are based on the concept of linear transformation, can not provide a unified powerful tool for nonlinear computational and analysis task. It is expected to introduce new matrix techniques to this field.

1.2. Introduction to the DQ, HDQ and DC Methods

The DQ method was introduced by the late Richard Bellman and his associates (Bellman et al., 1971, 1972) in the early 70s. and, since then, the technique has been applied in biosciences (Bellman et al., 1974b; Kashef et al., 1974), system identification (Bellman et al., 1974a, b, 1979; Hu et al, 1974), diffusion (Mingle, 1977), transport process (Civan et al., 1983a), fluid dynamics (Shu et al., 1992a, b, 1994a, b; Civan, 1993), chemical engineering (Quan et al., 1989; Wang, 1982; Chang, 1993; Civan, 1994b), lubrication (Malik et al., 1994), acoustics (Gutierrez et al., 1994) and Contact problems (Malik et al., 1993), etc. Recently, the differential quadrature method are extensively used to analyze deflection, vibration and buckling of linear and nonlinear structural components by Bert's group (Bert et al., 1988a, 1989, 1993, 1994a, b, c, 1996a; Jang et al., 1989; Striz et al., 1988; Farsa et al., 1991; Kukretic et al., 1992; Kang et al., 1995, 1996; Malik, 1996) and other researchers (Laura et al., 1993, 1994a, b, 1996; Pandy et al., 1991; Sherbourne, 1991; Du, 1994; Liew, 1994; Lin, 1994). Many scholars have made important contributions to this method and its applications. Mingle (1973) proposed a linear transformations in the DQ method to simplify the computing effort of the evaluation of the DQ weighting coefficients for high order derivatives. Civan et. al (1984b) extended this method to multi-dimensional problems and integro-differential equations. Bert et. al (1988) first applied this method successfully to structural component analysis. Qian et. al (1989a) gave the explicit formulas to compute accurately and efficiently the DQ weighting coefficients for the 1st and 2nd order derivative, and pointed out that the DQ method is actually equal to the collocation method. Shu et al. (1992b) applied the DQ method to fluid dynamic problems using parallel computation based on the multi-domain technique and gave a general recursion formulas for computing the DQ weighting coefficients. Wang and Bert (1993a) proposed a new and efficient approach in applying the DQ method to high order boundary value problems. Chang et al. (1993) employed the proper functions as basic functions instead of polynomial functions in the DQ method for dealing with problems involving steep gradients successfully, while Bert et al. (1993) and Striz et al. (1995) developed the harmonic differential quadrature (HDQ) method, which uses harmonic functions instead of polynomial as test function in the quadrature method to handle with periodic problems efficiently, and also circumvented the limitation for the number of grid point in the conventional DQ method based on polynomial test function. Their study showed that the proper test functions are essential for the computational efficiency and reliability of the DQ method. Striz et. al. (1994a) gave a domain decomposition technique in applying the DQ method to truss and frame structures successfully. Based on the cubature rule for multidimensional numerical integration (Engels, 1980), Civan (1989, 1994a) developed the DQ method to propose the differential cubature (DC) method as an competitive numerical technique for the solution of multi-dimensional differential and integro-differential equations. At present the studies on the DC method are few. But this method is very attractive to practical engineering computations. The DQ method has been also included in some books (Bellman et al., 1973, 1985, 1986, Zwillinger, 1992)

Though the differential quadrature method has been applied in many areas, the method does not yet attract extensive attentions in proportional to its simplicity and high efficiency. The ensuing section introduces the notation of the differential quadrature, harmonic differential quadrature and differential cubature methods.

1.2.1. Differential Quadrature and Harmonic Differential Methods

The essence of the DQ method is that the partial derivative of a function with respect to a variable is approximated by a weighted sum of function values at all discrete points in that direction. Its weighting coefficients do not relate to any special problem and only depend on the grid spacing. Thus, any partial differential equation can be easily reduced to a set of algebraic equations using these coefficients. Considering a function f(x) with N discrete grid points (Civan et al., 1984b), we have

$$\frac{\partial^m f(x)}{\partial x^m}\Big|_{x_i} = \sum_{j=1}^{N} w_{ij}^{(m)} f(x_j), \qquad i = 1, 2, \ldots, N. \qquad (1.2\text{-}1)$$

where x_j 's are the discrete points in the variable domain. $f(x_j)$ and $w_{ij}^{(m)}$ are the function values at these points and the related weighting coefficients, respectively. In order to determine the weighting coefficients $w_{ij}^{(m)}$, equation (1.2-1) must be exact for all polynomials of degree less than or equal to (N-1).

To avoid the ill-conditioning the Vandermonde matrix in the calculation of the weighting coefficients, the Lagrange interpolation basis functions (Quan et al., 1989a; Shu et al., 1992a, b; Bert et al., 1993; Chen et al., 1993a) are used as the test functions, namely,

$$f_k(x) = \prod_{i \neq k}^{N} \frac{x - x_i}{x_k - x_i}. \qquad (1.2\text{-}2)$$

Substituting Eq. (1.2-2) into Eq. (1.2-1) yields the following two formulae to compute directly the weighting coefficient of the 1st order derivative (Quan and Chang, 1989a), i.e.,

$$A_{ij} = \frac{1}{x_j - x_i} \prod_{\substack{k \neq i,j \\ i \neq j}}^{N} \frac{x_i - x_k}{x_j - x_k}, \qquad i = 1,2,\ldots,N \text{ and } j = 1,2,\ldots,N. \qquad (1.2\text{-}3a)$$

and

$$A_{ii} = -\sum_{k \neq i}^{N} \frac{1}{x_i - x_k}, \qquad i = 1,2,\ldots,N. \qquad (1.2\text{-}3b)$$

For higher order derivatives, the weighting coefficients can be generated by one recursion formula (Shu and Richards, 1992b)

$$w_{ij}^{(m+1)} = m\left(A_{ij} w_{ii}^{(m)} - \frac{w_{ij}^{(m)}}{x_i - x_j}\right), \qquad i \neq j \qquad (1.2\text{-}4)$$

and

$$w_{ii}^{(m+1)} = -\sum_{j \neq i}^{N} w_{ij}^{(m+1)}, \qquad (1.2\text{-}5)$$

where the superscript (m) and (m+1) denote the order of the derivative.

The harmonic differential quadrature method is a new development of the differential quadrature method, which has been used successfully to solve a variety of problems. The HDQ method chooses harmonic functions as its test functions instead of polynomials in the DQ method, i.e.,

$$f(x) = \{1, \sin\pi x, \cos\pi x, \sin 2\pi x, \cos 2\pi x, \ldots, \sin(N-1)\pi x/2, \cos(N-1)\pi x/2\}, \qquad (1.2\text{-}6)$$

where N is an odd number. Substituting Eq. (1.2-6) into Eq. (1.2-1), we obtain N sets of N order algebraic equations. Solving these linear algebraic equations, we can obtain the weighting coefficients of the HDQ method. Using the same idea for obtaining the formulas (1.2-4) and (1.2-5), Wang (1995) gave the direct computing formulas for computing weighting coefficients of the HDQ method. The HDQ method has been found especially efficient for problems with periodic behaviors (Bert et al., 1993; Striz et al., 1995).

The weighting coefficients for high order derivatives in the DQ and HDQ methods can also be obtained by matrix multiplication (Mingle, 1973; Civan, 1984b). For example, considering the DQ weighting coefficients A_{ij}, B_{ij}, C_{ij} and D_{ij} which are corresponding to the first, second, third and fourth order derivatives, respectively, we have

$$B_{ij} = \sum_{k=1}^{N} A_{ik} A_{kj}, \qquad C_{ij} = \sum_{k=1}^{N} A_{ik} B_{kj}, \qquad D_{ij} = \sum_{k=1}^{N} B_{ik} B_{kj}, \qquad (1.2\text{-}7)$$
$$i, j = 1,2,\ldots,N.$$

1.2.2. Differential Cubature Method

The DC method is different from the DQ method in that a partial derivative of the function with respect to a coordinate direction is expressed as a weighted sum of the function values at all discrete points in the entire multi-dimensional solution domain rather than simply in that coordinate direction. Considering a two-variable function f(x, y) (Civan; 1989, 1994a), we obtains the partial derivatives with respect to x and y expressed as

$$\frac{\partial^m f(x, y)}{\partial x^m}\Big|_i = \sum_{j=1}^{N} w_{ij}^{(xm)} f_j, \qquad i = 1, 2, \ldots, N. \tag{1.2-8}$$

and

$$\frac{\partial^p f(x, y)}{\partial y^p}\Big|_k = \sum_{j=1}^{N} w_{kj}^{(yp)} f_j, \qquad k = 1, 2, \ldots, N. \tag{1.2-9}$$

where j (or i, k) denotes the one-dimensional indexing of the two-dimensional grid points and f_j is the function value at the corresponding grid point. $w_{ij}^{(xm)}$ and $w_{ij}^{(yp)}$ are the DC weighting coefficients for the related partial derivatives. Note that the number of grid points N in the Eqs. (1.2-8) and (1.2-9) is in the entire multi-dimensional domain rather than only in a coordinate direction as in Eq. (1.2-1) of the DQ method. The DC method degenerates into the DQ method in one dimensional problems. Similar to Eq. (1.2-7) of the DQ method, it is straightforward that there exists the following formulas for the DC weighting coefficient matrix E_x, E_y, F_x, F_x and F_{xy} for the 1st and 2nd order derivatives, i.e.,

$$F_x = E_x^{\,2}, \qquad F_y = E_y^{\,2}, \qquad F_{xy} = E_x E_y = F_y E_x. \tag{1.2-10}$$

For higher order derivatives, similar formulas exist. As was pointed out by Malik and Civan (1994a), the major time-consuming calculations in the DC method are to compute the weighting coefficients. The present formulas can effectively reduce these efforts.

More detailed descriptions on the DQ-type methods see Bellman, Kashef and Casti (1972), Civan and Sliepcevich (1984b), Qian and Chang (1989a), Shu and Richards (1992b), Malik and Civan (1994a), Bert and Malik (1996d), etc.

1.3. Thesis Overview

This paper focuses on the differential quadrature method and relative nonlinear computations. Our work is a step forward in nearly all important basic aspects, including the truncation error, the choice of sampling points, the algebraic structures of the weighting coefficient matrices, numerical stability, and new approximate formulas in matrix form for applying the DQ method to the initial- or/and boundary value problems. The paper is the first authentic attempt to use the DQ method for initial value problem. We also introduce two kinds of special matrix product to nonlinear computations and analysis of the DQ method as well as other numerical methods. We hope to present a new framework for nonlinear computation and analysis of general purpose. In the following we briefly state our main contributions.

First, the conventional formulas of the truncation error in the DQ method, derived by Bellman et al. (1972) by using Rolle’s theorem, do not involve the practical grid interval and are too imprecise for many practical purpose. We gave new truncation error formulas based on the polynomial interpolation technique. Applying our formulas, the truncation error at any grid point can be estimated much more accurately, and it was also found that the DQ method is high order accurate numerical method. It is also noted that the Chebyshev grid points are not optimal grid spacings although it may be the most efficient in all existing grid spacings for some cases. It is known that the Chebyshev polynomials are derived from the optimal polynomial approximation of a function rather than of the derivatives of a function. The DQ truncation error constants around the ends using such grid spacing are obviously larger than at other grid points nearby the center region. Therefore, such approximation do not satisfy the optimal minimax approximation principle. Similar situations also occur in the DQ method using the equally spaced grid points or zeros of Legendre polynomials. Through the numerical trial and error, we propose seven general rules for choosing grid spacings. The numerical experiments have also demonstrated that the DQ method using the grid spacings which satisfies these rules has faster rate of convergence than using the zeros of the Chebyshev polynomials. Since the DQ method is in fact equivalent to the collocation and pseudo-spectral methods, the presented truncation error formulas and rules for choosing sampling points are also applicable for these two methods. We also give the simplified formulas for computing the DQ weighting coefficients under equally spaced grid points and the zeros of the Chebyshev polynomials or Legendre polynomials and overcome the difficulty that the conventional applications of zeros of orthogonal polynomials can not include the boundary points.

Second, we gave the DQ approximate formulas in matrix form for boundary value problems instead of traditional polynomial ones. By using these formulas, the formulation effort is greatly reduced and the fast algorithms in the solution of Lyapunov matrix equation are introduced to the DQ calculations of multi-dimensional boundary value problems of the second order, and the computational effort is reduced by about N^3 times. The high efficiency and simplicity of the presented reduction computation were demonstrated in the DQ solution of the Poisson equation and steady convection-diffusion equations. We notice the fact that the rank of the DQ weighting coefficient matrices for the ith order derivative is M-i, where M is the number of grid points. Moreover, the coefficient matrix is in fact power zero matrix, namely the weighting coefficient matrix is zero matrix when M=i. Therefore, the coefficient matrices in the DQ method must be modified into full rank matrices before practical computation. Based on these facts, we proposed a new approach in applying the multiple boundary conditions in the DQ solution of boundary value problems of more than two order, which are often encountered in structural engineering analysis. The approach is employed to analyze the static bending, vibration and buckling of plates and beams. Compared with the other existing approaches, the presented approach shows easy use, good stability, wide applicability and high accuracy.

Third, the stiff ordinary differential equations are often encountered in automatic control, electronic network, biosciences, physics and chemical kinetics. The traditional numerical integration methods are not applicable for such problems. The key to handle stiff problems is the numerical stability of methods. On the other hand, the structural dynamic analysis is a fundamental task in civil engineering, geomechanics, ocean platform and mechanical design and production. The problems of such type are also in general “stiff”. We first applied the DQ method to the initial value problems and pointed out that the method is A-stable. Therefore, the DQ method is always stable for stiff problems. Since the damping yields the negative real eigenvalues in general structural dynamic analysis, the DQ method is unconditionally stable for such problems. We presented the DQ approximate formulas in matrix form for initial value problems. Thus, the resulting formulations are Lyapunov algebraic matrix equation. Several fast algorithms for the solution of the Lyapunov matrix equations are successfully applied for the present purpose, and the computational effort and storage requirements are alleviated by an order of N^3 and N^2, respectively, where N is the number of interior grid points. Therefore, the DQ method requires nearly the same computational effort as the other single step methods such as Newmark, Wilson θ, Houblot, Runge-Kutta and Gear methods, etc. while maintaining its high computational efficiency. Some obstacles need be overcome to apply the DQ method for boundary value problems with complex geometries. However, the difficulties are not inherent in applying the DQ method to the initial value problems. So the initial value problems may be a very significant application of the DQ method in practice.

Fourth, the symmetric grid spacings such as equally spaced grid points and the zeros of orthogonal polynomial (for example, the Chebyshev polynomial and Legendre polynomial, etc.) are often employed in the DQ and HDQ methods. We proved that in such situations the weighting coefficient matrices of the DQ and HDQ methods are centrosymmetric or skew centrosymmetric ones. The centrosymmetric matrix can be factorized into tow smaller size matrices in the evaluation of its inverse, determinant, eigenvectors and eigenvalues. Therefore, the computational effort and storage requirements in applying the DQ method for certain problems (especially for problems with symmetric boundary conditions) can be reduced by 75 per cent and 50 per cent, respectively. The skew centrosymmetric matrix is first studied and found to have the similar factorization properties of the centrosymmetric matrix. The practical applications in the DQ solutions of steady-state convection-diffusion problems and static deflection, vibration, buckling of beams and plates also showed that the computational effort and storage requirements are greatly reduced and the scale of computed problems is enlarged.

Fifth, it may be our most important contributions that the special matrix products are introduced to the nonlinear computations of the DQ methods as well as other numerical methods, including finite element, finite difference, boundary element, collocation, pseudo-spectral, spectral, Newmark, Wilson θ, Runge-Kutta and Gear methods, etc. By using the Hadamard product, we obtained two new types of nonlinear formulation for nonlinear computations of general purpose, which are denoted as the formulation-S in the ordinary and Kronecker product form and the formulation-H in the Hadamard product form. The nonlinear formulation efforts are greatly reduced and programming task is simplified. It is known that the stability analysis of nonlinear computations is not an easy task. We derived explicit formulas to describe the relation between the condition number and perturbed error bound for the nonlinear formulation-S and formulation-H. The idea of instability and ill-conditioning are closely related in numerical computations. Therefore, the present formulas establish the basis for stability and error analysis of nonlinear computations. The computational efficiency of the standard iteration formula in the simple iteration method is usually not high. This is reason why the simple iteration method is not so popular as the Newton-Raphson method. Due to applying the Hadamard power and function concepts, the construction of the efficient iterative formulas in the simple iteration method becomes similar to handling nonlinear scalar function and very easy to be accomplished. Thus, the Hadamard product approach offers a compact, efficient and convenient procedure for use of the simple iteration method. The convergence speed is linear in the simple iteration method and, by contrast, quadratical in the Newton-Raphson method. However, the Newton-Raphson method requires computing the Jacobian derivative matrix and its inverse. The Newton-like iterative methods also suffer from the serious practical disadvantages in that these iterations will converge only if suitable initial values can be found. The simple iteration method does not require so stringent initial guess. For these reasons, the simple iteration method becomes very competitive to the popular Newton-Raphson methods as well as its variants. The conclusion is also demonstrated via numerical examples. The most important feature of the Hadamard product may be its ability to express the nonlinear relation in the problems of interest.

We first defined a new special product of matrix and vector, SJT[*] product. The SJT product provides a very simple and highly efficient approach to accurate calculation of the Jacobian derivative matrix in the Newton-Raphson method for the solution of the nonlinear formulations. Due to the utility of the Jacobian matrix in a wide range of science and engineering areas, the SJT product may have high potential for many theory and applied analysis. Also, the coupling nonlinear formulations for the corresponding coupling nonlinear partial differential equations can in general be decoupled by means of the Hadmard product and SJT product. Therefore, the computational effort and storage requirements are alleviated extremely. The Hadamard product and SJT product approach was successfully applied in the DQ analysis of geometrically nonlinear bending of orthotropic plates and nonlinear vibration of beams as well as fluid dynamics problems. For geometrically nonlinear bending analysis, the computational effort and storage requirements are reduced to about only one twenty-seventh and one-ninth, respectively, as much as those by Bert et al. due to decoupling computations.

We also give a new approximate formula for directly computing the inverse of the Jacobian matrix in the Newton-Raphson method. The formulas only involves the ordinary matrix multiplications and does not require the conventional inversion computation as in the Gauss elimination method or the Gauss-Jordan method. Therefore, the possible ill-conditioning of the Jacobian matrix in the iteration process, which often occurs in the solution of large nonlinear systems, can not affect the convergence of the Newton-Raphson method. By converting some nonlinear differential operators into a combination of a linear operator and a simpler nonlinear operator, the cross nonlinear algebraic terms in the DQ and DC formulation for problems of interest are eliminated or reduced and thus the requirements for virtual memory and computational effort are reduced greatly. Finally, several critical problems for further developing the DQ method and Hadamard product and SJT product techniques into a engineering approaches are also discussed.

In appendix I, we discuss the algebraic and analysis properties of the Hadamard product. These properties may be valuable in a wide range of nonlinear computation and analysis. In appendix II, we also proved that the weighting coefficient matrices of the quadrature method based on Fourier-type trigonometric principle are circulant ones.

It is stressed that the following chapters place their emphasis on our recent work. The details on the related work by other researchers can be found in an excellent survey provided by Bert and Malik (1996d), and not presented here in great detail for the sake of brevity.

[*] SJT is the abbreviate of Shanghai Jiao Tong University

CHAPTER 2.
SOME BASIC ASPECTS ON NONLINEAR COMPUTATIONS

2.1. Introduction

Matrix computations is of vital importance in nonlinear analysis and computations. The traditional linear algebraic approach, which is very successful for linear numerical computations, has been extended to handle the nonlinear problems. However, since nonlinear problems have actually different from linear ones, linear algebraic and the relative matrix approaches, which are based on the concept of linear transformation, can not provide a unified powerful tool for nonlinear computational and analysis task. In fact, the ordinary matrix product computation seems not to undertake the task of nonlinear analysis and computations very well. We need seek an alternate matrix approach to handle nonlinear problems. Recently, contributions was made by the present authors to solve this problem (Chen et al., 1996c,e,f). The Hadamard product of matrices was introduced to nonlinear numerical computations of the differential quadrature, differential cubature methods and other numerical techniques successfully. The Hadamard product is a kind of very simple special matrix computations and not well known to the numerical computation community. By using the Hadamard product, the nonlinear formulation effort of the DQ and DC methods are greatly reduced, and the formulations can be expressed in explicit and easily programmable matrix form. We also first defined the SJT product, a new kind of product of matrix and vector, to efficiently obtain analytical solution of the Jacobian derivative matrix in the Newton-Raphson method for the solution of the nonlinear formulations in the Hadamard product form (Chen and Zhong, 1996c).

The nonlinear formulation of the FD, collocation and pseudo-spectral methods can be concisely expressed in the Hadamard product form. The formulations in the Hadamard product form are denoted as formulation-H in this paper. By using the Hadamard product and SJT product, a new type of nonlinear formulation in the ordinary and Kronecker product form can be obtained in the FD, DQ, DC, pseudo-spectral, finite element, boundary element, spectral, least square, Galerkin methods, Runge-Kutta, Wilson θ, Newmark, and Gear methods, etc. The formulation of such type is represented by the formulation-S in the latter discussion.

We further develop the foregoing work in the following areas. First, the unified approaches are proposed to compute accurately and efficiently the Jacobian derivative matrix in the Newton-Raphson method for the solution of general formulation-S. It is worth pointing out that the Jacobian matrix of the formulation-H can be efficiently computed by using the SJT product approach no matter what numerical methods are used. Second, for the linear equations, the condition number measures the effect of the round-off on the resulting solutions of equations. We extend the approach to the error analysis of the nonlinear formulation-S and formulation-H using the norm properties of the Hadamard product. The relation between the perturbed error bound and the condition number is clearly expressed in the presented formulas. Third, Chen et al. (1996c, f) found that the DQ formulation-H for nonlinear coupling differential equations can be easily decoupled by using Hadamard and SJT product. Therefore, the computational effort and complexity are reduced greatly. We introduce the relative Jacobian matrix concept to simplify the decoupling computation further. Geometrically nonlinear bending of beams and plates under a uniformly distributed loading is investigated and the decoupling computation is tested successfully. Finally, the simple iteration method is applied in the solution of the formulation-H. Due to applying the Hadamard power, the construction of the efficient iterative formulas in the simple iteration method becomes very easy to be accomplished. Therefore, the simple iteration method become very competitive to the Newton-Raphson method. In addition, we give a new approximate formula for computing the inverse of the Jacobian matrix so as to circumvent the possible ill-conditioning of the Jacobian matrix. A simple approach is also presented to simplify the formulations of some nonlinear operators (Chen and Zhong, 1996c).

2. 2. Hadamard Product

We introduce the notation of the Hadamard product of matrices and state its some properties first (Ni, 1984; Horn, 1990). Based on the Hadamard product concept, the Hadamard power and function are also defined (Horn, 1990). Horn (1990) gave an excellent survey on the Hadamard product and its some applications.

Definition 2.2.1 Let matrices $A=[a_{ij}]$ and $B=[b_{ij}]\in C^{N\times M}$, the Hadamard product of matrices is defined as $A°B= [a_{ij}\, b_{ij}]\in C^{N\times M}$. where $C^{N\times M}$ denotes the set of N×M real matrices.

Definition 2.2.2 If matrix $A=[a_{ij}]\in C^{N\times M}$, then $A^{°q}=[a_{ij}^{q}]\in C^{N\times M}$ is defined as the Hadamard power of matrix A, where q is a real number. Especially, if $a_{ij} \neq 0$, $A^{°(-1)}=[1/a_{ij}]\in C^{N\times M}$ is defined as the Hadamard inverse of matrix A. $A^{°0}=11$ is defined as the Hadamard unit matrix in which all elements are equal to unity.

Definition 2.2.3 If matrix $A=[a_{ij}]\in C^{N\times M}$, then the Hadamard matrix function $f^{\circ}\left(A\right)$ is defined as $f^{\circ}\left(A\right)=\left[f\left(a_{ij}\right)\right]\in C^{N\times M}$.

Theorem 2.2.1: letting A, B and $C\in C^{N\times M}$, then

1> $A°B=B°A$ (2-1a)

2> $k(A°B)=(kA)°B$, where k is a scalar. (2-1b)

3> $(A+B)°C=A°C+B°C$ (2-1c)

4> $A°B=E_N^T(A\otimes B)E_M$, where matrix E_N (or E_M) is defined as $E_N =[\, e_1\otimes e_1 \vdots \cdots \vdots e_N\otimes e_N]$, $e_i=[0\cdots 0 \underset{i}{1} 0\cdots 0]$, $i=1, \cdots, N$, E_N^T is the transpose matrix of E_N. ⊗ denotes the Kronecker product of matrices. (2-1d)

5> If A and B are non-negative, then

$\lambda_{\min}\left(A\right)\min\left\{b_{ii}\right\} \le \lambda_{j}\left(A\circ B\right) \le \lambda_{\max}\left(A\right)\max\left\{b_{ii}\right\}$, where λ is the eigenvalue. (2-1e)

6> $(\det A)(\det B)\le\det(A°B)$, where det( ) denotes the determinant. (2-1f)

In appendix A we will further discuss the algebraic and analysis properties of the Hadamard product in greater detail

In the following discussions, we assume that the related boundary conditions for all given examples have been applied to the DQ weighting coefficient matrices using the approach proposed by Wang and Bert (1993a) or these boundary conditions are substituted directly into the DQ and DC weighting coefficient matrices (Civan and Sliepcevich, 1984b) Therefore, the boundary conditions are no longer considered separately. It should be noted that the modified DQ weighting coefficient matrices $\overline{A}_x$, $\overline{A}_y$, $\overline{B}_x$ and $\overline{B}_y$ for multi-dimensional problems here are different from the $\overline{A}_x$, $\overline{A}_y$, $\overline{B}_x$ and $\overline{B}_y$ defined in Wang et al. (1993a 1994a) in that the present ones are the resulting and stacked DQ weighting coefficient matrices in a multi-dimensional domain, while the latter are the DQ weighting coefficient matrices only in one dimensional sense.

Considering the quadratic nonlinear differential operator $\frac{\partial f(x,y)}{\partial x}\frac{\partial^2 f(x,y)}{\partial y^2}$, its DQ formulation can be expressed in Hadamard product form as

$$\frac{\partial f(x,y)}{\partial x}\frac{\partial^2 f(x,y)}{\partial y^2} = \left(\bar{\mathrm{A}}_{\mathrm{x}}\vec{f}\right)\circ\left(\bar{B}_y\vec{f}\right) \tag{2.2-2}$$

The formulation (2.2-2) has explicit matrix form. The DQ formulations in Hadamard product form for some nonlinear differential operators often encountered in practice can be expressed in a similar way:

1. $c(x,y)U_{,x} = \left\{c\left(x_{j,}y_j\right)\right\}\circ\left(\bar{A}_x\vec{U}\right)$, (2.2-3a)
2. $\left(U_{,x}\right)^q = \left(\bar{A}_x\vec{U}\right)^{\circ q}$, where q is a real number, (2.2-3b)
3. $\frac{\partial U^m}{\partial x}\frac{\partial U^n}{\partial y} = \left(\bar{A}_x\vec{U}^{\circ m}\right)\circ\left(\bar{A}_y\vec{U}^{\circ n}\right)$, (2.2-3c)
4. $\sin U_{,x} = \sin^{\circ}\left(\bar{A}_x\vec{U}\right)$, (2.2-3d)
5. $\exp\left(U_{,x}\right) = \exp^{\circ}\left(\bar{A}_x\vec{U}\right)$, (2.2-3e)

where $\bar{B}_{xy}$, $\bar{A}_x$ and $\bar{A}_y$ denote the DQ weighting coefficient matrices for the corresponding partial derivatives, modified by the related boundary conditions, respectively.

Striz et al. (1994a) analyzed the driven cavity problem by the differential quadrature method. The DQ formulation was given in matrix form in Eq. (15) on p. 668 in Striz et al (1994a), namely

$$\begin{aligned}\{F(\psi)\} = [A_y]\{\psi\}[A_x]\left([B_x]+[B_y]\right)\{\psi\} - [A_x]\{\psi\}[A_y]\left([B_x]+[B_y]\right)\{\psi\} - \\ \frac{1}{\mathrm{Re}}\left([D_x]+2[B_x]+2[B_x][B_y]+[B_y]+[D_y]\right)\{\psi\} = \{0\}\end{aligned} \tag{2.2-4}$$

where $[D_x]$, $[D_y]$, $[B_x]$, $[B_y]$, $[A_x]$ and $[A_y]$ are the corresponding DQ weighting coefficient matrices for the first, second and fourth order derivative along the x- and y- directions, respectively. The details see the reference. The nonlinear terms in the formulation are incorrectly stated, since the ordinary matrix product was used to formulate the nonlinear operator. It is impossible to express the nonlinear formulation of the DQ method in the matrix form without the Hadamard product. In fact, equation (2.2-4) can be not run because the operation conditions for the ordinary matrix multiplication are not satisfied obviously.

The correct matrix formulation for the nonlinear differential equation must use the Hadamard product and is expressed as

$$\begin{aligned}\{F(\psi)\} = \left([A_y]\{\psi\}\right)\circ\left([A_x]\left([B_x]+[B_y]\right)\{\psi\}\right) - \left([A_x]\{\psi\}\right)\circ\left([A_y]\left([B_x]+\right.\right. \\ \left.\left.[B_y]\right)\{\psi\}\right) - \frac{1}{Re}\left([D_x]+2[B_x]+2[B_x][B_y]+[B_y]+[D_y]\right)\{\psi\} = \{0\}\end{aligned} \tag{2.2-5}$$

Feng and Bert (1992) applied the differential quadrature method to analyze the geometrically nonlinear vibrations of beams. In the present study, due to applying the Hadamard product of matrices, the nonlinear formulation is greatly simplified. A explicit matrix formulation is obtained. Therefore, programming effort is reduced.

The governing equation for this case can be normalized as

$$\frac{d^4 v}{d\,\xi^4} - \frac{3}{8}\frac{a^2}{r^2}\left(\int_0^1 \left(\frac{dv}{d\xi}\right)^2 d\xi\right)\frac{d^2 v}{d\,\xi^2} - \varpi^2 v = 0\,, \tag{2.2-6}$$

where $r^2 = \frac{I}{A}$, I is centroidal moment of inertia of beam, A is area of beam cross section, a is amplitude, v is nonlinear normal mode; ξ=x/L, L is length of beam, x is axial position coordinate; and $\varpi^2=\omega^2 mL^4/EI$ is the dimensionless frequency, ω is nonlinear frequency, m is mass per unit length, E is modulus of elasticity. For more details see Feng and Bert (1992).

The formulation for equation (2.2-6) can be expressed in Hadamard product form as

$$\overline{D}\vec{V} - \frac{3}{8}\frac{a^2}{r^2}\left\{\vec{G}\left[\left(A\tilde{V}\right)\circ\left(A\tilde{V}\right)\right]\right\}\overline{B}\vec{V} - \varpi^2\vec{V} = 0 \tag{2.2-7}$$

where $\overline{B}$ and $\overline{D}$ are the DQ weighting coefficient matrices, modified by the respective boundary conditions for the 2nd and 4th order derivatives, respectively. The order of matrices $\overline{B}$ and $\overline{D}$ varies for the various techniques in applying the multiple boundary conditions, namely, n-2 for Wang and Bert's new technique and n-4 for the so-called δ technique (Jang et al. 1989; Bert et al., 1996d) or new technique proposed by Chen et al. (1993b, 1994) and Du et al. (1994). Here n is the number of grid points. Since the boundary conditions have been used in the formulation of $\overline{B}$ and $\overline{D}$, they are no longer considered. $\vec{V}$ is a (n-2)×1 mode vector at inner grid points. $\tilde{V} = \left\{0, \quad \vec{V}^T, \quad 0\right\}^T$, A is original n×n DQ weighting coefficient matrix for the 1st order derivative. Since the upper and lower bounds of the integral in Eq. (2.2-7) are constants, it is not necessary to utilizes the DQ method for numerical integration as in Feng and Bert (1992). We herein use the Newton-Cotes numerical integration approach for simplicity. $\vec{G}$ is a 1×n vector composed of the Cotes coefficients for numerical integration. It is observed that the DQ formulation equation (2.2-7) has an explicit, compact and simple matrix form, and is obviously easier for programming than the conventional one expressed in a polynomial form in Feng and Bert paper. Also, it is noted that there is an typographical errors in the nonlinear formulation equation (16) in Feng and Bert (1992), namely, the first operation in that equation should be minus rather than plus.

2.3. SJT Product and its Some Properties and Applications

The Newton-Raphson method is a standard numerical technique to solve the nonlinear equation set resulting from the DQ (or the other numerical methods) solution of the nonlinear differential or integro-differential equations. One of the major time-consuming calculation in the Newton-Raphson method is to compute the Jacobian derivative matrix. In this section, we will provide an efficient and explicit procedure to compute the analytical solution of the Jacobian matrix of the nonlinear formulation in Hadamard product form. First, we herein present a new multiplication operation – SJT product of matrix and vector:

Definition 2.3.1 If matrix $A=[a_{ij}]\in C^{N\times M}$, vector $U=\{u_j\}\in C^{N\times 1}$, then $A\Diamond U=[a_{ij}\,u_j]\in C^{N\times M}$ is defined as the postmultiplying SJT product of matrix A and vector U, where ◊ represents the SJT product. If M=1, A◊B=A°B.

Definition 2.3.2 If matrix A=[a_{ij}]∈$C^{N\times M}$, vector V={v_j}∈$C^{M\times 1}$, then $V^T\lozenge A=[a_{ij}\,v_i]\in C^{N\times M}$ is defined as the SJT premultiplying product of matrix A and vector V.

2.3.1. SJT product for the evaluation of Jacobian matrix of formulation-H

The iteration formula of the Newton-Raphson method in solution of the nonlinear algebraic equations is

$$\vec{U}^{(k+1)} = \vec{U}^{(k)} - \left[\frac{\partial\varphi\left(\vec{U}^{(k)}\right)}{\partial\vec{U}}\right]^{-1}\varphi\left(\vec{U}^{(k)}\right), \tag{2.3-1}$$

where $\frac{\partial}{\partial\vec{U}}$ denotes the Jacobian derivative matrix operator, $\vec{U}$ denotes the desired vector, $\varphi(\vec{U})=\left\{\varphi_1(\vec{U}) \quad \varphi_2(\vec{U}) \quad \dots \quad \varphi_m(\vec{U})\right\}^T$ is nonlinear algebraic equations. The corresponding Jacobian derivative matrix is defined as

$$\frac{\partial\varphi\{\vec{U}\}}{\partial\vec{U}} = \begin{bmatrix} \frac{\partial\varphi_1}{\partial U_1} & \frac{\partial\varphi_1}{\partial U_2} & \cdots & \frac{\partial\varphi_1}{\partial U_m} \\ \frac{\partial\varphi_2}{\partial U_1} & \frac{\partial\varphi_2}{\partial U_2} & \cdots & \frac{\partial\varphi_2}{\partial U_m} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial\varphi_m}{\partial U_1} & \frac{\partial\varphi_m}{\partial U_2} & \cdots & \frac{\partial\varphi_m}{\partial U_m} \end{bmatrix}. \tag{2.3-2}$$

Considering the nonlinear DQ formulation (2.2-2) in section 2.2, its Jacobian matrices can be obtained by

$$\frac{\partial}{\partial\vec{U}}\left\{\left(\overline{A}_x\vec{f}\right)\circ\left(\overline{B}_y\vec{f}\right)\right\} = \overline{A}_x\lozenge\left(\overline{B}_y\vec{f}\right) + \overline{B}_y\lozenge\left(\overline{A}_x\vec{f}\right). \tag{2.3-3}$$

Eq. (2.3-3) gives the accurate solutions for the Jacobian matrix of analog approximate term in Eq. (2.2-2) through simple algebraic computations, and computational effort is reduced greatly. The SJT premultiplying product are related to the Jacobian matrix for the DQ formulations such as $\frac{dU^m}{dx} = \overline{A}\vec{U}^m$, i.e.,

$$\frac{\partial}{\partial\vec{U}}\left\{\overline{A}_x\vec{U}^m\right\} = \left(m\vec{U}^{\circ(m-1)}\right)^T\lozenge\overline{A}_x. \tag{2.3-4}$$

In the following, we discuss some operation rules in applying the SJT product to compute the Jacobian matrix for the nonlinear formulation (2.2-3) given in section 2.2.

1. $$\frac{\partial}{\partial\vec{U}}\left\{\left\{c\left(x_{j,}y_j\right)\right\}\circ\left(\overline{A}_x\vec{U}\right)\right\} = \overline{A}_x\lozenge\left\{c\left(x_{j,}y_j\right)\right\} \tag{2.3-5a}$$

2. $$\frac{\partial}{\partial\vec{U}}\left\{\left(\overline{A}_x\vec{U}\right)^{\circ q}\right\} = q\overline{A}_x\lozenge\left(\overline{A}_x\vec{U}\right)^{\circ(q-1)}. \tag{2.3-5b}$$

3. $$\frac{\partial}{\partial\vec{U}}\left\{\left(\overline{A}_x\vec{U}^{\circ m}\right)\circ\left(\overline{A}_y\vec{U}^{\circ n}\right)\right\} = m\left\{\left(\vec{U}^{\circ(m-1)}\right)\lozenge\overline{A}_x\right\}\lozenge\left(\overline{A}_y\vec{U}^{\circ n}\right) + n\left\{\left(\vec{U}^{\circ(n-1)}\right)\lozenge\overline{A}_y\right\}\lozenge\left(\overline{A}_x\vec{U}^{\circ m}\right) \tag{2.3-5c}$$

4. $$\frac{\partial}{\partial\vec{U}}\left\{\sin\left(\overline{A}_x\vec{U}\right)\right\} = \overline{A}_x\lozenge\cos^{\circ}\left(\overline{A}_x\vec{U}\right) \tag{2.3-5d}$$

5. $$\frac{\partial}{\partial\vec{U}}\left\{\exp^{\circ}\left(\overline{A}_x\vec{U}\right)\right\} = \overline{A}_x\lozenge\exp^{\circ}\left(\overline{A}_x\vec{U}\right) \tag{2.3-5e}$$

6. If $\vec{y} = f^{\circ}(\vec{g})$, $\vec{g} = \varphi^{\circ}(\vec{U})$, we have $\dfrac{\partial \vec{y}}{\partial \vec{U}} = \dfrac{\partial \vec{y}}{\partial \vec{g}} \dfrac{\partial \vec{g}}{\partial \vec{U}}$. (2.3-5f)

In the above equations $(\ )_{,x} = \partial(\ )/\partial x$, $\dfrac{\partial}{\partial \vec{g}}$ and $\dfrac{\partial}{\partial \vec{U}}$ represent the Jacobian derivative matrix of certain Hadamard function with respect to vectors $\vec{g}$ and $\vec{U}$, respectively. It is observed from these formulas that the Jacobian derivative matrix for the nonlinear formulation-H can be computed by using the SJT product in the chain rules similar to those in differentiation of a scalar function. The above computing formulas give the analytical solutions of the Jacobian matrix for the problems considered. The computational effort for a SJT product is only n^2 scalar multiplications, namely, each entry in the Jacobian matrix is obtained by one scalar multiplication, which may be the minimal computing cost. It is emphasized that the SJT product approach can be efficiently implemented in a parallel treatment way.

The finite difference method is usually employed to obtain the approximate solutions of the Jacobian matrix in practice and requires $O(n^2)$ scalar multiplications. Therefore, both the SJT product approach and the finite difference method are essentially comparable in computing effort. However, the numerical approximate Jacobian matrix yielded by the finite difference method often affects the accuracy and convergence rate of the Newton-Raphson method. In contrast, the SJT product produces the analytic solution of the Jacobian matrix.

2.3.2. The algebraic properties of the SJT product

We notice the following fact that the SJT product is closely related with the ordinary product of matrices, namely,

If matrix $A=[a_{ij}]\in C^{N\times M}$, vector $\vec{U}=\{u_j\}\in C^{N\times 1}$, then the postmultiplying SJT product of matrix A and vector $\vec{U}$ satisfies

$A\Diamond\vec{U}=\mathrm{diag}\{u_1, u_2, \ldots\ldots, u_N\}A$, (2.3-6)

where matrix $\mathrm{diag}\{u_1, u_2, \ldots\ldots, u_N\}\in C^{N\times N}$ is a diagonal matrix whose main diagonal entries are the respective entries of vector $\vec{U}$. Similarly, for the SJT premultiplying product, we have

$\vec{V}^{T}\Diamond A = A\mathrm{diag}\{v_1, v_2, \ldots\ldots, v_M\}$, (2.3-7)

where vector $\vec{V}=\{v_j\}\in C^{M\times 1}$.

Based on Eqs. (2.3-4) and (2.3-5), we obtain the following properties of the SJT product, namely, if matrices A, B $\in C^{N\times M}$, vector $\vec{U}=[u_j]\in C^{N\times 1}$, $\vec{V}=[v_j]\in C^{M\times 1}$, then

1. $k_1k_2(A\Diamond\vec{U})=(k_1A)\Diamond(k_2\vec{U})=(k_2A)\Diamond(k_1\vec{U})$, where k_1 and k_2 are scalar. (2.3-8a)
2. $\vec{V}^{T}\Diamond(A\Diamond\vec{U})=(\vec{V}^{T}\Diamond A)\Diamond\vec{U}$ (2.3-8b)
3. $(A+B)\Diamond\vec{U}=A\Diamond\vec{U}+B\Diamond\vec{U}$ (2.3-8c)
4. $(A\circ B)\Diamond\vec{U}=A\circ(B\Diamond\vec{U})$ (2.3-8d)
5. $(A\Diamond\vec{U})^{T}=\vec{U}^{T}\Diamond A^{T}$, if n=m. (2.3-8e)
6. $(\vec{V}^{T}\Diamond A)^{T}=A^{T}\Diamond\vec{V}$, if n=m. (2.3-8f)
7. $(A\Diamond\vec{U})^{-1}=(\vec{U}^{\circ(-1)})^{T}\Diamond A^{-1}$, if $u_j\neq 0$ and A^{-1} exists. (2.3-8g)
8. $(\vec{V}^{T}\Diamond A)^{-1}=A^{-1}\Diamond\vec{V}^{\circ(-1)}$, if $v_j\neq 0$ and A^{-1} exists (2.3-8h)
9. $\det(A\Diamond\vec{U})=\det(A)\prod_{j=1}^{N}u_j$ (2.3-8i)

10. tr(A$\Diamond\vec{U}$)=$\sum_{j=1}^{N} a_{jj} u_j$, (2.3-8j)

11. $\left\| A\Diamond\vec{U} \right\|_\mu \ \le \ \left\| A \right\|_\mu \left\| \vec{U} \right\|_\mu$, (2.3-8k)

where $\| \ \|_\mu$ denotes the vector norm and the corresponding consistent submultiplicative matrix norm. The proofs for the properties are straightforward and omitted here for brevity.

2.4. Some Examples

In order to demonstrate the simplicity and efficiency of the Hadamard product and SJT product approach, we provide the following three detailed numerical examples.

Example 1: $y'' + \frac{1}{y} + \frac{y'^2}{y} = 0; \qquad y(0) = 1, y(1) = 2$. (2.4-1)

Considering the boundary conditions, the DQ formulation for the second order derivative of function y can be stated as

$$\begin{Bmatrix} y''_2 \\ y''_3 \\ \vdots \\ y''_{N-2} \\ y''_{N-1} \end{Bmatrix} = \begin{bmatrix} B_{22} & B_{23} & \cdots & B_{2,\,N-1} \\ B_{22} & B_{33} & \cdots & B_{3,\,N-1} \\ \vdots & \vdots & \vdots & \vdots \\ B_{N-2,\,2} & B_{N-2,\,2} & \cdots & B_{N-2,\,N-1} \\ B_{N-1,\,2} & B_{N-1,\,2} & \cdots & B_{N-1,\,N-1} \end{bmatrix} \begin{Bmatrix} y_2 \\ y_3 \\ \vdots \\ y_{N-2} \\ y_{N-1} \end{Bmatrix} + \begin{Bmatrix} B_{21} + 2B_{2,\,N} \\ B_{31} + 2B_{3,\,N} \\ \vdots \\ B_{N-2,\,1} + 2B_{N-2,\,N} \\ B_{N-1,\,1} + 2B_{N-1,\,N} \end{Bmatrix}. \qquad (2.4\text{-}2)$$

$$= \ \bar{B}\vec{y} \ + \ \vec{b}$$

Similarly, we have

$$\vec{y}\,' \ = \ \bar{A}\vec{y} \ + \ \vec{a} \,. \qquad (2.4\text{-}3)$$

The formulation for differential equation (2.4-1) in the Hadamard product form is stated as

$$\psi(\vec{y}) \ = \ \vec{y} \ \circ \left(\bar{B}\vec{y} \ + \ \vec{b}\right) \ + \ 1 \ + \ \left(\bar{A}\vec{y} \ + \ \vec{a}\right)^{\circ 2} \ = \ 0. \qquad (2.4\text{-}4)$$

By using the SJT product, we have

$$\frac{\partial\psi\{\vec{y}\}}{\partial\vec{y}} \ = \ I\Diamond\left(\bar{B}\vec{y} \ + \ \vec{b}\right) \ + \ \bar{B}\Diamond\vec{y} \ + \ 2\bar{A}\Diamond\left(\bar{A}\vec{y} \ + \ \vec{a}\right). \qquad (2.4\text{-}5)$$

The solutions for the linear differential equation

$$y'' = 0; \qquad y(0) = 1, y(1) = 2 \qquad (2.4\text{-}6)$$

are adopted as the initial guess values of the present Newton-Raphson iterative scheme. We obtains the convergence results with four iterations. The convergence criteria is the maximum absolute residual of Eq. (2.4-4) is less than or equal to 10^{-10}. The maximum relative error of the DQ results is less than 0.001 when using six Chebyshev points. It is observed that Eqs. (2.4-4) and (2.4-5) give explicit matrix form for the nonlinear computation of this example. Thus, the application of the DQ method is simplified. It should be noticed that the computational effort for each SJT product in Eq. (2.4-5) is only 16 multiplications.

Example 2. $y'' + \sin(y') + 1 = 0; \qquad y(0) = 0, y(1) = 1$ (2.4-7)

The DQ formulation for this differential equation can be expressed in the Hadamard matrix function form

$$\psi(\vec{y}) \ = \ \bar{B}\vec{y} \ + \ \vec{b} \ + \ sin^{\circ}\left(\bar{A}\vec{y} \ + \ \vec{a}\right) \ + \ 1 \ = \ 0, \qquad (2.4\text{-}8)$$

where $\bar{A}$, $\bar{B}$, $\vec{a}$ and $\vec{b}$ are obtained in the way similar to the above example. The Jacobian derivative matrix can be computed by

$$\frac{\partial\psi(\vec{y})}{\partial\vec{y}} = \bar{B}\vec{y} + \bar{A}\lozenge\, cos^{\circ}\left(\bar{A}\vec{y} + \vec{a}\right). \tag{2.4-9}$$

The solutions for linear differential equation

$$y'' + 1 = 0; \qquad y(0) = 0, y(1) = 1 \tag{2.4-10}$$

are chosen as the initial guess for the iteration. We obtain convergence results no more than four iterations when the maximum absolute residual of equations (2.4-8) is less than or equal to 10^{-10}. The results in two successive iterates agree to six significant digits.

As is shown in these two examples, the DQ method is very easy to be used for nonlinear problems by using the present Hadamard matrix function, Hadamard product, and SJT product approaches.

Example 3. $y' - e^{-y} = 0;$ y(0)=0, 0≤x≤1 (2.4-11)

The DQ formulation for this case is expressed as

$$\varphi(\vec{y}) = \bar{A}\vec{y} - e^{\circ(-\vec{y})} = 0 \tag{2.4-12}$$

The Jacobi matrix is given by

$$\psi\{\varphi\} = \bar{A} + I\lozenge e^{\circ(-\vec{y})} \tag{2.4-13}$$

Fast convergence and excellent results are again achieved by using the Newton-Raphson method as in the examples 1 and 2. Six Chebyshev grid points is used as sampling points, the relative error is no more than 0.001 with three iterations. In sections 4.5 and 5.6 there are several more complex examples using the Hadamard product and SJT product.

2.5. Decoupling Computations

It is a difficult task to compute the nonlinear coupling differential equations. Chen and Zhong (1996c) pointed out the decoupling computation in the DQ method by using the Hadamard product and SJT product. In this section we will introduce concept of the relative Jacobian derivative matrix among different dependent variables to simplify decoupling computations, and discussions are given on how to apply the Hadamard product and SJT product to decouple the nonlinear coupling formulation-H's. The following example will help to illustrate and clarify the present technique.

The example is geometrically nonlinear bending of the simply-supported beam under uniformly distributed loading. The normalized governing equations are given by

$$\frac{d^4W}{dX^4} - \left(\frac{1}{r\,l}\frac{dU}{dX} + \frac{1}{2}\left(\frac{dW}{dX}\right)^2\right)\frac{d^2W}{dX^2} = \frac{qL^4}{EI\,r} \tag{2.5-1a}$$

$$\frac{1}{r\,L}\frac{d^2U}{dX^2} + \frac{dW}{dX}\frac{d^2W}{dX^2} = 0, \tag{2.5-1b}$$

where A=area of beam cross section, E=modulus of elasticity, I=centroidal moment of inertia of beam cross section. The variables have been normalized in the form:

$$X = \frac{x}{L},\ r^2 = \frac{I}{A},\ W = \frac{w}{r},\ U = \frac{u}{r}. \tag{2.5-2}$$

In terms of the DQ method, we have the formulation-H:

$$D_w\vec{W} + \left[\frac{1}{rL}A_u\vec{U} - \frac{1}{2}\left(A_w\vec{W}\right)^{\circ 2}\right]\circ\left(B_w\vec{W}\right) = \frac{qL^4}{rEI} \tag{2.5-3a}$$

$$\frac{1}{r\,L}\,B_u\vec{U} \;+\; \left(A_w\vec{W}\right) \circ \left(B_w\vec{W}\right) \;=\; 0, \tag{2.5-3b}$$

where A$_u$ and B$_u$ denote the DQ weighting coefficient matrices for the 1st and 2nd order derivative of function U(x); A$_w$, B$_w$ and D$_W$ for the 1st, 2nd and 4th order derivatives of function W(x). Note that the related boundary conditions have been applied to these weighting coefficient matrices by means of the technique proposed by Wang and Bert (1993a). The vector $\vec{W}$ is chosen as the basic variable in the present computation. From Eq. (2.5-3b), we have

$$\vec{U} \;=\; -r\,LB_u\left(A_w\vec{W}\right) \circ \left(B_w\vec{W}\right) \tag{2.5-4a}$$

$$\frac{\partial\vec{U}}{\partial\vec{W}} \;=\; -r\,L\left(A_u\Diamond\left(B_w\vec{W}\right) \;+\; B_u\Diamond\left(A_w\vec{W}\right)\right), \tag{2.5-4b}$$

where

$$\frac{\partial\vec{U}}{\partial\vec{W}} = \begin{bmatrix} \frac{\partial U_1}{\partial W_1} & \frac{\partial U_1}{\partial W_2} & \cdots & \frac{\partial U_1}{\partial W_n} \\ \frac{\partial U_2}{\partial W_1} & \frac{\partial U_2}{\partial W_2} & \cdots & \frac{\partial U_2}{\partial W_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial U_n}{\partial W_1} & \frac{\partial U_n}{\partial W_2} & \cdots & \frac{\partial U_n}{\partial W_n} \end{bmatrix} \tag{2.5-5}$$

is the Jacobian derivative matrix of vector $\vec{U}$ with respect to vector $\vec{W}$. The variable vector $\vec{U}$ and its Jacobian derivative matrix can be obtained from variable vector $\vec{W}$ by using the above equations. The introduction of the relative Jacobian derivative matrix $\frac{\partial\vec{U}}{\partial\vec{W}}$ is a key idea to simplify decoupling computations, especially for equations with many coupled dependent variables. We choose equation (2.5-3a) as the basic equation, namely,

$$\phi\left(\vec{W}\right) \;=\; D_w\vec{W} \;+\; \left[\frac{1}{r\,L}\,A_u\vec{U} \;+\; \frac{1}{2}\left(A_w\vec{W}\right)^{\circ 2}\right] \circ \left(B_w\vec{W}\right) \;-\; \frac{qL^4}{r\,EI} \tag{2.5-6a}$$

批注 [JM1]:

$$\frac{\partial\phi}{\partial\vec{W}} = D_w + \left[\frac{1}{rL}\,A_u\,\frac{\partial\vec{U}}{\partial\vec{W}} + A_w\Diamond\left(A_w\vec{W}\right)\right]\Diamond\left(B_w\vec{W}\right) + B_w\Diamond\left[\frac{1}{rL}\,A_u\vec{U} + \left(A_w\vec{W}\right)^{\circ 2}\right]. \tag{2.5-6b}$$

The Newton-Raphson iteration equation for this case is

$$\vec{W}^{\,(k+1)} = \vec{W}^{\,(k)} - \left[\frac{\partial\phi\left(\vec{W}^{\,k}\right)}{\partial\vec{W}}\right]^{-1}\phi\left(\vec{W}^{\,(k)}\right). \tag{2.5-7}$$

The numerical results are obtained by using eleven Chebyshev grid points. The solutions for linear simply-supported beam are used as the iterative initial guess. Table 2-I displays the iteration number, linear and nonlinear solutions, and relative derivation for various loading q, where relative derivation is defined as (Nonlinear - Linear)/Nonlinear.

As can be observed, the iterative times increases as the loading increases. Compared with the results of linear modeling, the deflection W is decreased due to the consideration of the nonlinearity under large deflection situation. The results agree well with the theoretical analysis for this problems (Wang, 1969; Houlden, 1973). But the analytical solutions for the governing equation (2.5-1a, b) are not available in the literature. Note that the number of nonlinear algebraic equations are reduced from 2N×2N to N×N by means of decoupling. Therefore, the computational effort and storage requirements for this case are reduced by about 87.5% and 75%, respectively. It is also observed that the Newton-Raphson method still converge even if the initial linear solutions diverge greatly from the resulting nonlinear results. It is concluded that the formulation-H for nonlinear simultaneous partial differential equations can be easily decoupled by using the Hadamard product and SJT product techniques. More complex problems can also be decoupled in the same way to achieve the computational reduction, for example, geometrically nonlinear bending of plates which will discussed in later section 4.5.

Table 2-1. DQ solutions of a geometric nonlinear simply-supported beam under uniformly distributed loading

Loading (qL^4/EIr)	Iterative number	linear	Nonlinear	Relative deviation
1	1	0.0130208	0.0130202	-4.29E-5
5	2	0.06510	0.06503	-1.07E-3
10	2	0.13021	0.1297	-4.25E-3
30	3	0.3906	0.3771	-0.03596
50	3	0.6510	0.5972	-0.09024
100	4	1.302	1.027	-0.2675
150	5	1.953	1.341	-0.4566
200	5	2.604	1.587	-0.6408
300	6	3.906	1.966	-0.9865
500	7	6.510	2.501	-1.603
1000	8	13.021	3.445	-2.893
5000	12	65.104	6.0597	-9.744

2.6. On Simple Iteration Method

The Newton-Raphson method may be the most important numerical technique for the solution of nonlinear algebraic equations (Ortega and Rheinboldt, 1970), but sometimes the simple iteration method, which is also named as successive substitution method (Finlayson, 1980) or Picard method (Wright, 1964), is a very useful technique for the same task due to its simplicity. The purpose is of this section to show its ease and efficiency in solving the nonlinear formulation-H by means of the simple iterative method. The concept of the Hadamard power, which is referred to in section 2.2, is used in the present work.

Consider the set of nonlinear algebraic equations

$$F(x) = 0 \tag{2.6-1}$$

where x denotes the unknown vector. Generally, by adding x to the left side of equation (2.6-1), we can obtain the standard iterative scheme:

$$x^{(k+1)} = x^{(k)} + \alpha F\left(x^{(k)}\right) \tag{2.6-2}$$

where the superscript (k) means the iterate number. α is the iteration control parameter. After the initial guess $x^{(0)}$ is given, equation can be iterated repeatedly to obtain $x^{(1)}$, $x^{(2)}$ until $x^{(n)}$, which satisfies the required convergence accuracy. However, it is noted that α often is chosen to be very small value to assure the convergence, which leads to the low efficiency in the simple iteration method. Therefore, the utility of the simple iteration method is limited. The iteration equation (2.6-2) is named as the standard iteration equation in the simple iteration method in the later discussion, since this iteration equation is easily obtained and widely used in practice. In general, it is a very difficult task to construct iteration equations of other forms in applying the simple iteration method. However, equation (2.6-2) may be not the most efficient iteration scheme for the problem of interest. So the construction of the iterative equation is a key to improve the efficiency of the simple iteration method. The present study focuses on this problem.

The Hadamard power and function provide a simple and effective approach to construction of the iteration formulas in the solution of the formulation-H. The following example can illustrate our idea more clearly. Considering differential equation

$$y''y + y'^2 + 1 = 0, \qquad y(0) = 1, \quad y(1) = 2, \tag{2.6-3}$$

its formulation-H using the DQ method is

$$\phi(\vec{y}) = \vec{y} \circ \left(\overline{B}\vec{y} + \vec{b}\right) + \left(\overline{A}\,\vec{y} + \vec{a}\right)^{\circ 2} + 1 = 0. \tag{2.6-4}$$

where $\vec{a}$, $\vec{b}$, $\overline{A}$ and $\vec{B}$ are defined as in equations (2.4-2) and (2.4-3). We can construct six iteration equations for formulation (2.6-4).

1. $\vec{y} = \vec{y} + \alpha\phi(\vec{y}) = \vec{y} + \alpha\vec{y} \circ \left(\overline{B}\vec{y} + \vec{b}\right) + \left(\overline{A}\,\vec{y} + \vec{a}\right)^{\circ 2} + 1$ (2.6-5a)
2. $\vec{y} = \left[-1 - \left(\overline{A}\vec{y} + \vec{a}\right)^{\circ 2}\right] \circ \left(\overline{B}\vec{y} + \vec{b}\right)^{\circ(-1)}$ (2.6-5b)
3. $\vec{y} = B^{-1}\left[\left(-1 - \left(\overline{A}\vec{y} + \vec{a}\right)^{\circ 2}\right) \circ y^{\circ(-1)} - \vec{b}\right]$ (2.6-5c)
4. $\vec{y} = \overline{A}^{-1}\left[\left(-1 - \vec{y} \circ \left(\vec{B}\vec{y} + \vec{b}\right)\right)^{\circ 0.5} - \vec{a}\right]$ (2.6-5d)
5. $\vec{y} = \dfrac{1}{2}\,\overline{A}^{-1}\left[\vec{a}^{\circ(-1)} \circ \left(-\vec{y} \circ \left(\vec{B}\vec{y} + \vec{b}\right) - 1 - \left(A\vec{y}\right)^{\circ 2} - \vec{a}^{\circ 2}\right)\right]$ (2.6-5e)
6. $\vec{y} = \overline{A}^{-1}\left[-\vec{y} \circ \left(\vec{B}\vec{y} + \vec{b}\right) - 1 - 2\left(A\vec{y}\right) \circ \vec{a} - \vec{a}^{\circ 2}\right]^{\circ 0.5}$ (2.6-5f)

Equation (2.6-5a) is the standard iteration equation for this case, other five iteration equations apply the Hadamard power. It is noted that the present procedure is very simple and similar to handling nonlinear scalar function. The solutions of linear equation

$$y'' = 0, \qquad y(0) = 1, \quad y(1) = 2 \tag{2.6-6}$$

are used as the initial guess. The iteration using equation (2.6-5d) and (2.6-5f) is terminated due to occurring square root of real negative. Eqs. (2.6-5b) and (2.6-5e) do not converge. Only Eqs. (2.6-5a) and (2.6-5c) is convergence, but it is noted that the equation (2.6-5a) converges only when α is less than 0.01. In order to compare, table 2-2 gives the number of iterations taken in the Newton-Raphson and the simple iteration method under the same convergence criterion. We choose here the maximum successive relative error as the convergence criterion, which is defined as

$$err(y_i) = \left|\frac{y_i^{(k+1)} - y_i^{(k)}}{y_i^{(k+1)}}\right|. \tag{2.6-7}$$

The conventional relative error, which is defined as the ratio of the absolute error to the absolute analytical solutions, may be not a good convergence criterion for the present purpose, since the accuracies of the numerical results depend not only on the iteration solution procedures but on the DQ method itself. In fact, the conventional relative errors in this case are basically invariable when the successive relative error is no less than 10^{-5}. Therefore, higher requirement for successive relative error is only to compare the convergence rate without other practical significance. Note that the solutions of equation (2.6-6) are also employed as the initial guess in the Newton-Raphson method and six Chebyshev grid points are used as sampling points.

As can be seen from table 2-2, there is not obvious difference in the convergence speed of the Newton-Raphson method and iteration equation (2.6-5c) under lower convergence criterion. However, the difference becomes apparent when very small successive relative error are required. The reason is that the convergence is linear in the simple iteration method and, by contrast, quadratical in the Newton-Raphson method. Although the Newton-Raphson method has faster speed of convergence, the method requires computing the Jacobian derivative matrix and its inverse. Therefore, as was pointed out by Finlayson (1980), the simple iteration method may be preferred if the method is effective for the considered cases. Based on the tradeoff between computational effort and accuracy, at least in this case the simple iteration method proves more efficient. On the other hand, we can observe that iteration equation (2.6-5a) has very slow speed of convergence. Thus, it is not practical.

The given example shows that, by using the Hadamard power, the construction of the iteration formulas in the simple iteration method becomes a very simple task for the solution of the formulation-H. The SJT product presented in section 2.3 can be applied to evaluate the Jacobian matrix of these iteration equations and determine which iteration equation may be the effective in the simple method in advance by using certain matrix norm.

Finding good starting guess is a difficult task either in the simple iteration method or in the Newton-Raphson method. However, the simple iteration method is semilocal convergence or even global convergence for convex function, while the Newton-like methods are always local convergence. Thus, the simple iteration methods has more large convergence domain than the latter, namely, the former is more easier to seek the initial guess. This is a very important factor considered in many practical uses. In addition, the Newton iteration equation in the Newton-Raphson method are sometimes ill-conditioning and thus affect the convergence of the solution procedure. For these reasons, the simple iteration method become very competitive to the Newton-Raphson method.

Table 2-2. Comparison on iterative numbers between the Newton-Raphson and simple iteration methods

Methods	Maximum successive relative error∠				
	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-11}
Newton-Raphson	3	4	4	4	5
Eq. (2.6-5c)	4	7	10	13	16
Eq. (2.6-5a)	20	111	214	317	420

2.7. Approximate Formulas for Computing Inverse of Jacobian Matrix

Systems of nonlinear algebraic equations are usually solved by the simple iteration and Newton-Raphson methods. In section 2.6 we reviewed these two methods in the solution of nonlinear formulations in Hadamard product form based on the criteria of computational efficiency, choice of starting guess and employing simplicity. The simple iteration method has the major advantage of simplicity in that no

Jacobian derivative matrix need be calculated and no matrix need be inverted. In section 2.3 the computing effort in the Jacobian evaluations has been minimized to O(n^2) multiplications by using the SJT product. On the other hand, the matrix inversion may take considerable computation time for large problem (e.g. $n^3/3$ multiplications and divisions) and even verge in some unfavorable situations towards impossibility due to ill-conditioning of large Newton iteration equations. To avoid the inversion computation, the simplified Newton-Raphson method and modified Newton-Raphson method have been well developed. However, the rate of convergence is affected in these variants of the Newton-Raphson method. The ill-conditioning of the Newton iteration equations is usually alleviated by introducing a damping factor, namely, so-called the Levenberg-Marguardt method. However, the choice of damping factor may not be an easy task. The trade-off among these Newton-like methods involves the number of iterations and the work per iteration. In this study, we hope to present a simple approximate formula for computing the inversion of the Jacobian matrix. Some numerical comparisons are also provided. The merits and drawbacks of the formula are discussed in the closing part.

First, we give the following two theorems (Cheng, 1989):

Theorem 2.7.1: If matrix A is convergence matrix, e.g. $\|A\| \prec 1$, then

$$(I - A)^{-1} = I + A + A^2 + \cdots + A^m + \cdots. \tag{2.7-1}$$

where I is unite matrix, $\| \ \|$ represents certain norm of matrix.

Theorem 2.7.2: If $\|A\| \prec 1$, then

$$\left\|(I - A)^{-1} - \left(I + A + A^2 + \cdots + A^m\right)\right\| \le \frac{\|A\|^{m+1}}{1 - \|A\|}, \tag{2.7-2}$$

where m is non negative integral. The theorem describes the error for an approximation of (I-A)$^{-1}$.

The Newton-Raphson iteration formula can be in general expressed as

$$\begin{cases} x_{k+1} = x_k - A_k^{-1} f\left(x_k\right) \\ A_{k+1} = A_k + \delta A_k \end{cases} \tag{2.7-3}$$

where A_k is the Jacobian matrix, the superscript k denotes the iterative number, f(x) is the nonlinear algebraic equations, x is the desired vector.

According to the above theorems 2.7.1 and 2.72, we have

$$A_{k+1}^{-1} = \left(A_k + \delta A_k\right)^{-1} \cong A_k^{-1} - A_k^{-1} \delta A_k A_k^{-1} \tag{2.7-4}$$

if $\left\|A_k^{-1} \delta A_k\right\| < 1$ and

$$\left\|\left(A_k + \delta A_k\right)^{-1} - \left(A_k^{-1} - A_k^{-1} \delta A_k A_k^{-1}\right)\right\| \le \frac{\left\|A_k^{-1}\right\| \left\|A_k^{-1} \delta A_k\right\|^2}{1 - \left\|A_k^{-1} \delta A_k\right\|}. \tag{2.7-5}$$

Applying the formula (2.7-4), we can directly computed inverse of the Jacobian matrix by using ordinary matrix product. In what follows we give several examples to demonstrate the effectiveness of the proposed formula.

Example 1. $$\begin{cases} 3x^2 + 4y^2 = 1 \\ y^3 - 8x^3 = 1 \end{cases} \qquad (2.7\text{-}6)$$

Example 2. $$\begin{cases} x_1 - 5x_2^2 + 7x_3^2 + 12 = 0 \\ 3x_1x_2 + x_1x_3 - 11x_1 = 0 \\ 2x_2x_3 + 40x_1 = 0 \end{cases} \qquad (2.7\text{-}7)$$

Example 3. $$yy'' + y'^2 + 1 = 0; \qquad y(0) = 1, \; y(1) = 2. \qquad (2.7\text{-}8)$$

Examples 1 and 2 are often used as numerical examples for the Newton-Raphson method in the standard numerical analysis or algorithms program textbooks, while example 3 is also used in section 2.4 and, thus, the detailed solution procedures are omitted here. These test problems are chosen just for simplicity and are not expected to be adequate in all circumstances.

The initial guess for example 1 is chosen as {-0.5, 0.25}. Under the same convergence criteria of max{$|f_i(x, y)|$}<10^{-7}, only two iterations converge in the Newton-Raphson method using either the present approximate formula or not. The simplified Newton-Raphson method take four iterations to converge. For example 2, the converge criteria requires the maximum residuals of equations less than 10^{-8}. The iteration number in the traditional, simplified, and present Newton-Raphson methods is 2, 4 and 7, respectively. Example 3 uses the same criteria as example 2 and six Chebyshev grid points are exploited as sampling points in the DQ method. The iteration number in three methods is 5, 6 and 11, respectively. According to these results, we can make the following conclusions.

At least the proposed approximate formula is applicable for the above examples. the convergence speed and accuracies of the Newton-Raphson method using the present approximate formula are affected to some extent. The simplified Newton-Raphson method has the slowest rate of convergence due to the fact that the method repeats using the same Jacobian matrix and its inverse. The reasonable choice of these methods in practice must consider the convergence speed, stability, and computational effort. The computing effort using the approximate formula is fairly higher in comparison to the conventional inversion computations such as the Gauss elimination or Gauss-Jordan methods since the present approach involves the twice matrix multiplications, but it does provide the salient significant features as stated in following three respects.

First, the approximate formula can circumvent effect of the possible ill-conditioning of the Newton iteration equations on the conventional inverse computation such as the Gauss elimination and Gauss-Jordan methods. Such situations are often encountered in practice. Second, It is known that the finite element and finite difference methods give rise to large sets of algebraic equations, and their solutions usually require inverting a matrix, or at least solving a large set of equations with a great many of zero elements. The solution of such equations, especially for large nonlinear systems, should utilize the zeros, otherwise, the computing cost will be unnecessarily very high. Therefore, the maintaining sparseness of coefficient matrix of such equations is of vital importance for the computational efficiency. The applications of the approximate formula can easily maintain the sparseness of coefficient matrix. Third, the Newton-Raphson method using the present approximate formulas involves a great deal of matrix multiplication, while the operation of ordinary matrix product is very well suited for employing parallel treatment. Therefore, the computing efficiency can be improved in the Newton-Raphson method using the approximate formula by means of the parallel computations. Based on these considerations, it is conclusions that the presented approximate formula may be useful for ill-conditioning, large sparse systems and especially suitable for parallel computations. The formulas may be very efficient when $\left\| A_k^{-1} \delta A_k \right\| << 1$.

From the standpoint of practical applications, the present study is rather preliminary. It is difficult to come to any general conclusion only from these examples, more experiences are required to determine the relative merits of the formula. The general applicability and efficiency of the formula is a current subject of further research.

2.8. An Approach for Simplifying Some Nonlinear Formulations

Following the idea of Quan et al. (1989a), Shu et al. (1992b), and Bert et al. (1993), we derive the DQ method from Lagrangian interpolation formula.

Let $W^2(x) = \sum_{j=1}^{N} \phi_j(x) \; W_j^{\,2}$, (2.8-1)

where $W_j = W(x_j)$, $\phi_j(x)$ are the Lagrangian interpolation basic functions.

$$\left(W_i^{\,2}\right)' = \frac{dW^2}{dx}\Big|_{x_i} = \sum_{j=1}^{N} \frac{\partial \phi_i(x_j)}{\partial x} W_j^2 = \sum_{j=1}^{N} A_{ij} W_j^2 , \tag{2.8-2}$$

where A_{ij} are the DQ weighting coefficients for the 1st order derivatives. Thus,

$$W_i W_i' = \frac{1}{2} \sum_{j=1}^{N} A_{ij} W_j^2 \ . \tag{2.8-3}$$

In matrix form, we have

$$\left\{ W_j W_j' \right\} = \frac{1}{2} [A] \left\{ W_j^{\,2} \right\} \tag{2.8-4}$$

Therefore, the nonlinear term ww' can be approximated by the DQ weighted sum of the square of function values at all discrete grid points. The conventional DQ approximation expression for ww' is

$$W_i W_i' = \left\{ \vec{W} \circ \left(A\vec{W} \right) \right\}_i = W_i \sum_{j=1}^{N} A_{ij} W_j \tag{2.8-5}$$

Obviously, Eq. (2.8-3) is a simpler formulation than Eq. (2.8-5) due to the elimination of cross nonlinear algebraic term $W_i W_j \left(i \neq j \right)$. Similarly, we have

$$\left\{ W_j^2 W' \right\} = \frac{1}{3} [A] \left\{ W_j^3 \right\} \tag{2.8-6}$$

$$\left\{ W_j'^2 + W_j W_j' \right\} = \frac{1}{2} [B] \left\{ W_j^{\,2} \right\} \tag{2.8-7}$$

where [B] denotes the DQ weighting coefficient matrix for the 2nd order derivatives. Generally, if the nonlinear operator NL{ } can be expressed as

$$NL\{ \ \} = L\{ \phi\{ \ \} \} \tag{2.8-8}$$

where L{ } is a linear operator, ϕ{ } is a nonlinear operator but simpler than NL{ }. The DQ or DC approximation of operator NL{ } can be given by

$$NL\left\{W(x)\right\}_i=\sum_{j=1}^{N}Q_{ij}\phi\left(W_j\right) \tag{2.8-9}$$

where Q_{ij}'s are the DQ weighting coefficients for linear operator L{ } and can been obtained by the sum of weighting coefficients of all single derivatives in linear operator L{ }. Since the nonlinear operator ϕ{ } is simpler than NL{ }, the corresponding DQ formulation is simpler and easier to be handled. It should be stressed that the boundary conditions in the present technique be converted into those only involving operator ϕ{W} before applying the approach, proposed by Wang and Bert (1993a), for implementing multiple boundary conditions. This conversion procedure is usually easy to be finished. If not using Wang and Bert's approach, the present DQ method can be used like the conventional DQ method, but the formulations are simplified greatly.

The nonlinear Boussinesq equations (Dodd et al., 1982) for shallow water waves can be stated as:

$$\begin{cases}\dfrac{\partial u}{\partial t}+u\dfrac{\partial u}{\partial x}+g\dfrac{\partial h}{\partial x}+\dfrac{1}{3}H\dfrac{\partial^3 u}{\partial t^2\partial x}=0\\ \dfrac{\partial h}{\partial t}+u\dfrac{\partial h}{\partial x}+h\dfrac{\partial u}{\partial x}=0\end{cases}. \tag{2.8-10}$$

Considering $u\dfrac{\partial u}{\partial x}=\dfrac{1}{2}\dfrac{\partial u^2}{\partial x}$, $u\dfrac{\partial h}{\partial x}+h\dfrac{\partial u}{\partial x}=\dfrac{\partial uh}{\partial x}$, the DQ formulation for Eq. (2.8-10) can be expressed as

$$\begin{cases}\left\{\dfrac{du_j}{dt}\right\}+\dfrac{1}{2}\left[\overline{A}_{u^2}\right]\left\{u_j^{\,2}\right\}+g\left[\overline{A}_h\right]\left\{h_j\right\}+\dfrac{1}{3}H\dfrac{d^2}{dt^2}\left(\left[\overline{A}_u\right]\left\{u_j\right\}\right)=0\\ \left\{\dfrac{dh_j}{dt}\right\}+\left[\overline{A}_{uh}\right]\left\{u_jh_j\right\}=0\end{cases}, \tag{2.8-11}$$

where g and H are constants. $\left[\overline{A}_{u^2}\right]$, $\left[\overline{A}_h\right]$, $\left[\overline{A}_u\right]$ and $\left[\overline{A}_{uh}\right]$ are the DQ weighting coefficient matrices modified by the boundary conditions for the corresponding function u^2, h, u and uh, respectively. It is observed that equation (2.8-11) eliminates the cross nonlinear algebraic terms in the conventional nonlinear formulations. Therefore, the requirements of virtual memory and computing effort are reduced greatly. This will be significant in practice, especially for some on-line computations. The following gives a numerical comparison in detail. Considering the differential equation (example 8 in Quan and Chang (1989)).

$$U_{,x}^2+UU_{,xx}+U_{,xx}=0,\qquad U(0)=0,U(1)=1. \tag{2.8-12}$$

First, by using the standard procedure, we have

$$\psi\left(\vec{U}\right)=\left(\overline{A}_u\vec{U}+\vec{a}\right)^{\circ 2}+\vec{U}\circ\left(\overline{B}_u\vec{U}+\vec{b}\right)+\overline{B}_u\vec{U}+\vec{b}=0. \tag{2.8-13}$$

Applying the Newton-Raphson method to solve the above nonlinear matrix equation, we can obtain the Jacobian derivative matrix by the following formula:

$$\frac{\partial\psi\left\{\vec{U}\right\}}{\partial\vec{U}}=2\overline{A}_u\Diamond\left(\overline{A}_u\vec{U}+\vec{a}\right)+\overline{B}_u\Diamond\vec{U}+I\Diamond\left(\overline{B}_u\vec{U}+\vec{b}\right)+\overline{B}_u. \tag{2.8-14}$$

In contrast, we applying the technique presented to this example. Eq. (2.8-12) can be simplified

$$\frac{1}{2}\frac{d^2U^2}{dx^2}+\frac{d^2U}{dx^2}=0. \tag{2.8-15}$$

The corresponding boundary conditions are

$$U(0) = 0, U(1) = 1 \quad and \quad U^2(0) = 0, \quad U^2(1) = 1. \qquad (2.8\text{-}16)$$

The second derivatives of functions U^2 and U can be approximated by the DQ method as

$$\frac{d^2\vec{U}^2}{dx^2} = \bar{B}_{u^2}\vec{U}^{\circ 2} + \vec{b}_{u^2}, \; and \quad \frac{d^2\vec{U}}{dx^2} = \bar{B}_u\vec{U} + \vec{b}_u. \qquad (2.8\text{-}17)$$

Generally, the modified DQ weighting coefficient matrices for functions U^2 and U are different, but for this case, $\bar{B}_{u^2}$ and $\bar{B}_u$ for function U^2 and U are the same due to the same boundary conditions by chance. We have the DQ formulation for Eq. (2.8-15)

$$\psi(\vec{U}) = \frac{1}{2}\bar{B}_{u^2}\vec{U}^{\circ 2} + \bar{B}_u\vec{U} + \vec{b}_{u^2} + \vec{b}_u = 0. \qquad (2.8\text{-}18)$$

The Jacobian derivative matrix can be given by

$$\frac{\partial\psi(\vec{U})}{\partial\vec{U}} = \vec{U}^T \Diamond \bar{B}_{u^2} + \bar{B}_U. \qquad (2.8\text{-}19)$$

Note that the premultiplying SJT product proposed in section 2.3 is used here. Eqs. (2.8-18) and (2.8-19) are obviously simpler than Eqs. (2.8-13) and (2.8-14).

The solutions for the linear differential equation

$$U_{,xx} = 0; \qquad U(0) = 0, U(1) = 1 \qquad (2.8\text{-}20)$$

is adopted as the starting guess values of the Newton-Raphson method. The Jacobian derivative matrix is computed by Eqs. (2.8-14) or (2.8-19), and the convergences are achieved with four iterations. The six grid points are used for this case. Results are displayed under the column e_u of table 2-3. The e_u is defined to be the relative errors in computation, i.e., the ratio of the absolute error to the absolute analytical solutions.

As is observed, the technique presented in this section obtains more accurate DQ results than the conventional procedure. In addition, the computing effort and storage requirements are also reduced greatly. In conclusion, a simple technique in the DQ method is presented to eliminate or reduce the cross nonlinear algebraic terms in the resulting DQ formulations for some differential operators and therefore computational efficiency of the DQ and DC methods can be improved significantly.

Table 2-3. Comparison of the Numerical Results with the Analytical Solutions

	Conventional approach	Present approach
	e_u	e_u
x_2	7.16E-4	5.93E-8
x_3	1.37E-4	3.60E-7
x_4	7.76E-5	2.36E-7
x_5	5.25E-5	3.30E-7

2.9. Nonlinear Computations of Other Numerical Methods

Based on the foregoing work, this section places its emphasis in the applications of the Hadamard product and SJT product techniques for general nonlinear numerical computations. The comparison of the advantages and disadvantages among a varity of the numerical methods is not the purpose of this study.

In section 2.9.1 the Hadamard product is extended to the nonlinear computations of the finite difference (FD), collocation and pseudo-spectral methods. The nonlinear formulation in the Hadamard product form for these numerical methods can be easily obtained and has a explicit matrix form. The SJT product is an effective approach to compute the Jacobian matrix for the formulation-H. For the nonlinear problems with varying parameters, we derive the general formulation-S in the ordinary and Kronecker product. Furthermore, the Hadamard product is also first applied to the nonlinear computations of the method of weighted residuals (MWR) as well as the least square, Galerkin, finite element (FE), boundary element (BE) and spectral methods. The same formulation-S in the ordinary and Kronecker product form is also obtained as in the foregoing pseudo-spectral method, etc. Due to special importance of the FE method in practice, more detailed discussion is also devoted to the FE method. Finally, some conclusions are obtained based on the present work. It is believed that the formulation-S is a standard formulation form for nonlinear numerical computations of general purpose.

In section 2.9.2 we will give direct computing formulas for the Jacobian matrix of the formulation-H and -S. Section 2.9.3 deals with the perturbed error analysis of the formulation-H and -S.

2.9.1. Formulation-H and Formulation-S

2.9.1.1. Nonlinear formulations in the finite difference, differential quadrature, differential cubature, collocation and pseudo-spectral methods

The work deals with the application of the Hadamard and SJT product for the general nonlinear numerical computations. The linear differential operator can be approximated by the finite difference, DQ and DC methods (Chen et al., 1996c, e, f) in the matrix form as

$$\left\{\frac{\partial^m W_j}{\partial x^m}\right\}_{n\times 1} = \left[D_x^{(m)}\right]_{n\times n}\left\{W_j\right\}_{n\times 1}, \tag{2.9-1}$$

where n is the number of interior discrete points. $D_x^{(m)}$ is the FD, DQ or DC coefficient matrices for the mth order derivatives. $\{W_j\}$ is the vector composed of unknown function values. Unlike the FD, DQ and DC methods, the pseudo-spectral and collocation (Canuto et al., 1988; Wright, 1964; Lapidus et al., 1985) methods use the spectral variables as the basic unknowns instead of the unknown function values. Therefore, a slightly different approximate formula is given for the collocation and pseudo-spectral method, namely,

$$\left\{\frac{\partial^m W_j}{\partial x^m}\right\}_{n\times 1} = \left[Q_x^{(m)}\right]_{n\times n}\left\{H_j\right\}_{n\times 1}, \tag{2.9-2}$$

where Q is the matrix representation of the collocation discretization of corresponding operator, H_j's are the unknown spectral vector. For simplicity, in the latter discussion we assume that the related boundary conditions have been directly incorporated into the above FD, pseudo-spectral, DQ and DC weighting coefficient matrices. Therefore, the boundary conditions are no longer considered separately.

We first consider quadratic nonlinear problems, the FD, DC and DQ formulations can be expressed in the Hadamard product form as

$$\frac{\partial^p W(x,y)}{\partial x^p}\frac{\partial^q W(x,y)}{\partial y^q} = \left(D_x^{(p)}\vec{W}\right)\circ\left(D_y^{(q)}\vec{W}\right). \tag{2.9-3}$$

Proof for Eq. (2.9-3) is straightforward. This is a key step in the latter analysis. Similarly, we have the formulation in the pseudo-spectral and collocation method:

$$\frac{\partial^p W(x,\ y)}{\partial x^p}\ \frac{\partial^q W(x,\ y)}{\partial y^q} = \left(Q_x^{(p)}\vec{H}\right) \circ \left(Q_y^{(q)}\vec{H}\right). \tag{2.9-4}$$

In the following we first discuss the FD, DQ and DC methods and the conclusions obtained are obviously applicable for the collocation and pseudo-spectral methods.

The FD, DC and DQ formulations for the linear and quadratic nonlinear differential operators with varying parameter can be stated as

$$c(x,\ y)\ \frac{\partial^p U(x,\ y)}{\partial x^p} = \left\{c\left(x_j,\ y_j\right)\right\} \circ \left(D_x^{(p)}\vec{U}\right) \tag{2.9-5a}$$

and

$$c(x,\ y)\ \frac{\partial^p U(x,\ y)}{\partial x^p}\ \frac{\partial^q U(x,\ y)}{\partial y^q} = \left\{c\left(x_j,\ y_j\right)\right\} \circ \left(D_x^{(p)}\vec{U}\right) \circ \left(D_y^{(q)}\vec{U}\right). \tag{2.9-5b}$$

The Kronecker product of matrices has the following property (Lancaster and Timenetsky, 1985)

$$(AB) \otimes (CD) = (A \otimes C)(B \otimes D). \tag{2.9-6}$$

Applying the above property, SJT product and equation (2.2-1d) of theorem 2.2.1 in section 2.2, we have

$$\left\{c\left(x_j,\ y_j\right)\right\} \circ \left(D_x^{(p)}\vec{U}\right) = \left(D_x^{(p)} \Diamond \left\{c\left(x_j,\ y_j\right)\right\}\right)\vec{U} \tag{2.9-7}$$

and

$$\left\{c\left(x_j, y_j\right)\right\} \circ \left(D_x^{(p)}\vec{U}\right) \circ \left(D_y^{(q)}\vec{U}\right) = \left\{c\left(x_j, y_j\right)\right\} \circ \left(E_n^T\left(D_x^{(p)}\vec{U}\right) \otimes \left(D_y^{(q)}\vec{U}\right)E_1\right) = \left\{c\left(x_j, y_j\right)\right\}$$
$$\circ\left(E_n^T\left(D_x^{(p)} \otimes D_y^{(q)}\right)\left(\vec{U} \otimes \vec{U}\right)E_1\right) = \left(\left(E_n^T\left(D_x^{(p)} \otimes D_y^{(q)}\right)\right)\Diamond\left\{c\left(x_j, y_j\right)\right\}\right)\left(\vec{U} \otimes \vec{U}\right) \tag{2.9-8}$$

where E₁ =1, n is the number of interior discrete points. Eqs. (2.9-7) and (2.9-8) isolate the FD or DQ coefficients from the unknown linear and nonlinear variables in the formulations.

For the general purpose, the quadratic nonlinear partial differential equation is given as

$$\sum_{k,l=0}^{N1} a_{kl} \frac{\partial^{(k+l)}U}{\partial x^k \partial y^l} + \sum_{k,l=0}^{N2} b_{kl}(x, y)\frac{\partial^{(k+l)}U}{\partial x^k \partial y^l} + \sum_{\substack{i,j=0\\k,l=0}}^{N3} c_{kl}(x, y)\frac{\partial^{(k+l)}U}{\partial x^k \partial y^l}\frac{\partial^{(i+j)}U}{\partial x^i \partial y^j} + C = 0 \tag{2.9-9}$$

where C is constant, a_{kl} is the constant coefficients. The above equation covers most of the quadratic nonlinear governing equations in practice.

The FD, DQ or DC formulation for the above differential equation can be expressed as

$$\sum_{k,l=0}^{N1} a_{kl} D_{xy}^{(k+l)}\vec{U} + \sum_{k,l=0}^{N2}\left\{b_{k,l}\left(x_j, y_j\right)\right\} \circ \left(D_{xy}^{(k+l)}\vec{U}\right) +$$
$$\sum_{\substack{i,j=0\\k,l=0}}^{N3}\left\{c_{k,l}\left(x_j, y_j\right)\right\} \circ \left(D_{xy}^{(i+j)}\vec{U}\right) \circ \left(D_{xy}^{(k+l)}\vec{U}\right) + \vec{F} = 0, \tag{2.9-10}$$

where $\vec{F}$ is constant vector. The formulation (2.9-10) belong to formulation-H form. Applying equations (2.9-7) and (2.9-8), we have

$$L_{n\times n}\vec{U} + Q_{n\times n^2}\left(\vec{U} \otimes \vec{U}\right) + \vec{F} = 0, \tag{2.9-11}$$

where

$$\begin{aligned} L_{n\times n} &= \sum_{k,l=0}^{N1} a_{kl} D_{xy}^{(k+l)} + \sum_{k,l=0}^{N2} D_{xy}^{(k+l)} \Diamond \left\{ b(x_j,\ y_j) \right\} \in C^{n\times n} \\ Q_{n\times n^2} &= \sum_{\substack{i,j=0\\k,l=0}}^{M} \left(E_n^{\,T} \left(D_{xy}^{(k+l)} \otimes D_{xy}^{(i+j)} \right) \right) \Diamond \left\{ c(x_j,\ y_j) \right\} \in C^{n\times n^2} . \end{aligned} \tag{2.9-12}$$

The above equation has formulation-S form. We can easily obtain formulation-H and formulation-S in the pseudo-spectral and collocation methods similar to Eqs. (2.9-10) and (2.9-11) in the same way, namely,

$$\begin{aligned} &\sum_{k,l=0}^{N1} a_{kl} Q_{xy}^{(k+l)} \vec{H} + \sum_{k,l=0}^{N2} \left\{ b_{k,l}\left(x_j,\ y_j\right) \right\} \circ \left(Q_{xy}^{(k+l)} \vec{H} \right) + \\ &\qquad \sum_{\substack{i,j=0\\k,l=0}}^{N3} \left\{ c_{k,l}\left(x_j,\ y_j\right) \right\} \circ \left(Q_{xy}^{(i+j)} \vec{H} \right) \circ \left(Q_{xy}^{(k+l)} \vec{H} \right) + C = 0 \end{aligned} \tag{2.9-13}$$

and

$$L_{n\times n} \vec{H} + Q_{n\times n^2} \left(\vec{H} \otimes \vec{H} \right) + \vec{F} = 0, \tag{2.9-14}$$

where the unknown spectral parameter vector $\vec{H}$ replaces the unknown function value vector $\vec{W}$ in Eqs (2.9-10) and (2.9-11). For the cubic nonlinear problems, we have similar general matrix formulation-S for the FD, DC and DQ methods:

$$L_{n\times n} \vec{U} + R_{n\times n^3} \left(\vec{U} \otimes \vec{U} \otimes \vec{U} \right) + \vec{F} = 0, \tag{2.9-15}$$

where $L \in C^{n\times n}$ and $R \in C^{n\times n^3}$. For higher order nonlinear problems, the formulations can be easily obtained in the same way.

As a example, we consider equations (2.5-1a, b) for geometrically nonlinear bending of beams given in section 2.5. After some simple manipulations of the formulations (2.5-3a, b) in Hadamard product form, we have

$$D_w \vec{W} + \left[A_u B_u^{-1} \left(A_w \vec{W} \right) \circ \left(B_w \vec{W} \right) - \frac{1}{2} \left(A_w \vec{W} \right) \circ \left(A_w \vec{W} \right) \right] \circ \left(B_w \vec{W} \right) = \frac{qL^4}{EI\,r}, \tag{2.9-16}$$

where $r=\sqrt{I/A}$, radius of gyration of beam cross section, W=w/r. A_u, B_u, A_w, B_w and D_W are defined as in section 2.5. Furthermore, we obtain the formulation in formulation-S form:

$$D_w \vec{W} + G_{n\times n^3} \left(\vec{W} \otimes \vec{W} \otimes \vec{W} \right) - \frac{qL^4}{EI\,r} = 0 \tag{2.9-17}$$

where $G = \dfrac{1}{r^2} E_n^T \left[\left(A_u B_u^{-1} E_n^T \left(A_w \otimes B_w \right) - \dfrac{1}{2} E_n^T \left(A_w \otimes A_w \right) \right) \otimes B_w \right]$. The problem is cubic nonlinearity. The solution of equation (2.9-17) is given in section 2.9.2.

It is noted that the foregoing nonlinear formulations are expressed either formulation-H (Eqs. (2.9-10) and (2.9-12)) or formulation-S (Eqs. (2.9-11), (2.9-13) and (2.9-14)), respectively. In fact, except for the FD method, other above-mentioned methods has very close relation with the collocation method (Bellman, 1973; Canuto and Hussaini, 1988; Chen and Zhong, 1996c). It is obvious that the formulation-H and formulation-S can be derived in the nonlinear computations of other numerical techniques in the family of the collocation method, for example, the discrete ordinate (DO) method (Shizgal et al., 1984; Mansell et al., 1993), spline DQ method (Bellman et al., 1975), and quadrature element method (QEM), which was introduced by Striz, et al (1994b) as a kind of the DQ finite element method.

2.9.1.2 Nonlinear formulations for the Galerkin, finite element, boundary element, spectral and least square methods

1. On the method of weighted residuals

It is known that the Galerkin, least square, finite element, boundary element and spectral methods can be derived form the method of weighted residuals (Lapidus and Pinder, 1982; Finlayson, 1972; Szabo and Lee, 1969; Huebner, 1975). Therefore, we first apply the Hadamard product to the nonlinear computation of the method of weighted residuals. In the MWR, the desired function ϕ in the differential governing equation

$$\psi\{\phi\} - f = 0 \tag{2.9-18}$$

is replaced by a finite series approximation $\hat{\phi}$,

$$\phi = \hat{\phi} = \sum_{i=1}^{N} C_i U_i \tag{2.9-19}$$

where $\psi\{\ \}$ is a differential operator. U_i can be defined as the assumed functions and C_j‘s are the unknown parameters. In the Galerkin, finite element and spectral methods, the function U_j are usually chosen to satisfy certain boundary conditions of considered problems and are also variously denoted as shape, trial or basis functions. The approximate function $\hat{\phi}$ is completely specified in terms of unknown parameters C_j. Substituting this approximation $\hat{\phi}$ into the governing equation (2.9-18), it is in general unlikely that the equation will be exactly satisfied, namely, result in a residual R

$$\psi\{\hat{\phi}\} - f = R \tag{2.9-20}$$

The method of weighted residuals seeks to determine the N unknowns C_j in such a way that the error R is minimized over the entire solution domain. This is accomplished by requiring that weighted average of the error vanishes over the solution domain. Choosing the weighting function W_j and setting the integral of R to zero:

$$\int_D \left[\psi\{\hat{\phi}\} - f\right] W_j dD = \int_D R W_j dD = 0, \qquad j=1,2,....,N. \tag{2.9-21}$$

Equation (2.9-21) can be used to obtain the N unknown coefficients. This equation generally describes the method of weighted residuals. In order to state our idea clearly, considering the following the two-dimensional linear and nonlinear differential operators with varying parameter:

$$L_1\{\phi\} = c(x, y)\frac{\partial^p \phi}{\partial x^p} \tag{2.9-22a}$$

$$L_2\{\phi\} = c(x, y)\frac{\partial^p \phi}{\partial x^p}\frac{\partial^q \phi}{\partial y^q}\ . \tag{2.9-22b}$$

Substitution of Eq. (2.9-19) into Eqs. (2.9-22a) and (2.9-22b) and applications of equation (2.2-1d) in the theorem 2.2.1 result in

$$L_1\{\hat{\phi}\} = c(x, y)\left\{\frac{\partial^p U_i}{\partial x^p}\right\}^T \vec{C} \tag{2.9-23a}$$

$$
\begin{gathered}
L_2\left(\hat{\phi}\right) = c(x,y)\left[\left\{\frac{\partial^p U_i}{\partial x^p}\right\}^T \vec{C}\right]\circ\left[\left\{\frac{\partial^q U_i}{\partial x^q}\right\}^T \vec{C}\right] = E_1^T c(x,y) \\
\left(\left\{\frac{\partial^p U_i}{\partial x^p}\right\}^T \otimes \left\{\frac{\partial^q U_i}{\partial x^q}\right\}^T\right) E_1\left(\vec{C}\otimes\vec{C}\right) = c(x,y)\left(\left\{\frac{\partial^p U_i}{\partial x^p}\right\}^T \otimes \left\{\frac{\partial^q U_i}{\partial x^q}\right\}^T\right)\left(\vec{C}\otimes\vec{C}\right)
\end{gathered}
\tag{2.9-23b}
$$

where $\vec{C}$ is vector composed of the unknown parameters, E_1=1. Substituting the Eqs. (2.9-23a, b) into Eq. (2.9-21), we have

$$\int_D L_1\left\{\hat{\phi}\right\} W_j dD = \left[\int_D c(x,y)\left\{\frac{\partial^p U_j}{\partial x^p}\right\}^T W_j dD\right]\vec{C} \tag{2.9-24a}$$

$$\int_D L_2\left(\hat{\phi}\right) W_j dD = \left[\int_D c(x,y)\left(\left\{\frac{\partial^p U_i}{\partial x^p}\right\}^T \otimes \left\{\frac{\partial^q U_i}{\partial x^q}\right\}^T\right) W_j dD\right]\left(\vec{C}\otimes\vec{C}\right) \tag{2.9-24b}$$

Applying the above results, we can easily obtain the formulation of the method of weighted residuals for the nonlinear differential equation (2.9-9), namely,

$$K_{n\times n}\vec{C} + G_{n\times n^2}\left(\vec{C}\otimes\vec{C}\right) + \vec{F} = 0 \tag{2.9-25}$$

where $\vec{F}$ is the constant vector.

$$K_{n\times n} = \int_D\left(\sum_{k,l=0}^{N1} a_{kl}\left\{\frac{\partial^{(k+l)} U_i}{\partial x^k \partial y^l}\right\}^T + \sum_{k,l=0}^{N2} b_{kl}(x,y)\left\{\frac{\partial^{(k+l)} U_i}{\partial x^k \partial y^l}\right\}^T\right) W_j dD \in C^{n\times n}$$

and

$$G_{n\times n^2} = \int_D\left(\sum_{\substack{k,l=0\\ i,j=0}}^{N1} c_{kl}(x,y)\left\{\frac{\partial^{(i+j)} U_p}{\partial x^i \partial y^j}\right\}^T \otimes \left\{\frac{\partial^{(k+l)} U_p}{\partial x^k \partial y^l}\right\}^T\right) W_j dD \in C^{n\times n^2}$$

represent the coefficient matrices corresponding to the linear and nonlinear operators, respectively. Equation (2.9-25) has the same formulaton-S form as equations (2.9-11) and (2.9-13). It is easily proved that, for the cubic nonlinear differential equations, the resulting MWR formulation-S is similar to Eq. (2.9-14). Note that the evaluation of the coefficient matrices K and G is very suitable to be finished in a parallel treatment way.

The choice of the weighting functions W_j in the method of weighted residuals depends on the form of the error distribution principle. The different weighted functions are related with different weighted residual techniques. The most famous and often used weighted residual techniques are known as the Galerkin and least square methods. The least square method uses the derivative of the residual function R with respect to the unknown coefficients as the weighting functions, while the weighting functions W_j in the Galerkin method are chosen to be the same as the approximating function U_j in Eq. (2.9-19). Among the family of the Galerkin methods, the spectral (Haltiner and Williams, 1980), boundary element and finite element methods are most commonly used in engineering practice. The difference between the spectral and finite (or boundary) element methods is to choose the different basis functions. The latter selects the spline functions as the basis functions, namely “local function” or “piece interpolation function” technique is used in the finite element method and the basis functions are defined in certain subregion and zero elsewhere, while the spectral method usually uses the basis functions of higher-order continuity exhibiting global support, which are also formally required to be member of a complete set of functions.

As are discussed above, the only difference among the above-mentioned numerical methods lies in the use of different weighting and basis functions in the method of weighted residuals. From the foregoing deduction on equation (2.9-25), it is noted that the formulation-S can be derived no matter what weighting and basis functions we use in the method of weighted residuals. Therefore, it is straightforward that, by using the Hadamard product, we can obtain the formulation-S for the nonlinear computations of these methods. Since the finite element method has the particular importance in practical engineering, in the following we will discuss applying the Hadamard product to the finite element method in greater detail.

The collocation method is the simplest numerical technique in the family of the method of weighted residuals (Lapidus and Pinder, 1982), in which weighting function w_j is chosen to be the Dirac delta, namely $w_j = \delta\left(x - x_j\right)$. Therefore, the Hadamard product is suitable for the nonlinear computations of the collocation methods. In the section 2.9.1.1 we directly introduce the Hadamard product to the DQ, DC, collocation and pseudo-spectral methods in a straightforward and intuitive way. Obviously, the formulation-S and formulation-H in the collocation method can be also derived from the method of weighted residuals by using the Hadamard prodcut.

2. On finite element method

In the preceding discussion on the method of weighted residuals, we assumed that we were dealing with the entire solution domain. In the following the domain is broken into elements first, the main focus is on an individual element. By analogy with equation (2.9-19) we define a local approximation valid for only one element at a time. From the Galerkin method (Szabo and Lee, 1969), the residual equations governing the behavior of an element are expressed as

$$\int_{D^{(e)}} \left[L\left(\hat{\phi}^{(e)}\right) - f^{(e)}\right] N_j^{(e)} dD^{(e)} = \int_{D^{(e)}} R^{(e)} N_j^{(e)} dD^{(e)} = 0, \quad \text{j=1,2,...,r} \tag{2.9-26}$$

where the functions $N_j^{(e)}$ are recognized as the interpolation functions over the element. The superscript (e) restricts the range to one element, and $\phi^{(e)}=[N^{(e)}]\{\phi\}^{(e)}$, $f^{(e)}$=forcing function defined over element (e), r=number of unknown parameters assigned to the element. We can obtain a set of equations like equations (2.9-26). The shape functions N_j should be chosen to guarantee the interelement continuity necessary for the assembly process. Similar to the procedure obtaining Eq. (2.9-25), we can derive the matrix formulation for the quadratic nonlinear problems from element equation (2.9-26), namely,

$$K_{r\times r}^{(e)} \vec{W}^{(e)} + G_{r\times r^2}^{(e)}\left(\vec{W}^{(e)} \otimes \vec{W}^{(e)}\right) = F^{(e)} \tag{2.9-27}$$

where $\vec{W}^{(e)}$ denotes the nodal value of the unknown field variable or its derivatives instead of the foregoing undermined parameters C_j in equation (2.9-25). Note that equation (2.9-27) is defined over the element and expresses the properties of the individual elements. $K_{r\times r}^{(e)}$ and $G_{r\times r}^{2\ (e)}$ are the linear and nonlinear element coefficient matrices, respectively. The computational effort for the evaluation of coefficient $G^{(e)}$ is r(r+1)/2 numerical integrations, and can be reduced by using the property of the considered problems such as symmetricity. The parallel computation is also very feasible and efficient to evaluate coefficient matrices $K^{(e)}$ and $G^{(e)}$. It is noted that $G^{(e)}$ is also a very sparse matrix like the matrix K.

Next, we assemble matrix equations (2.9-27) expressing the behavior of the elements and form the matrix equations expressing the behavior of the entire system. This work can be accomplished in the same routine procedure as in the linear problems and is usually done by computer. Also, the coordinate transformation, from the local to the global system or from the global to the local system, has to be conducted. The resulting matrix equation for the system has the same form as equations (2.9-27) in an individual element except that they contain many more terms because they include all nodes, i.e.,

$$K_{n\times n}\vec{W} + G_{n\times n^2}\left(\vec{W}\otimes\vec{W}\right) = \vec{F} \tag{2.9-28}$$

Eq. (2.9-28) must be modified to account for the boundary conditions of the problem before the system equation are ready for solution. These work can be accomplished in the procedures similar to those in the conventional linear finite element method. It is noted that the only difference during formulating process between the finite element and other Galerkin methods is that the former has an assembly operation while the latter has not. In contrast, the linear formulation of the finite element method is in general stated as

$$K_{n\times n}\vec{W} = \vec{F} \tag{2.9-29}$$

where K is the same as equation (2.9-28) and often known as the stiffness matrix in solid mechanics. Eq. (2.9-29) has not the nonlinear term $G_{n\times n^2}\left(\vec{W}\otimes\vec{W}\right)$ in Eq. (2.9-28). By analogy with the linear FE formulation the conventional nonlinear formulation in the finite element method is usually expressed as

$$K\left(\vec{W}\right)\vec{W} = \vec{F} \tag{2.9-30}$$

where $K\left(\vec{W}\right)$ is the function of unknowns $\vec{W}$. The evaluation of the coefficient matrix $K\left(\vec{W}\right)$ usually requires much strenuous work and lack the ease and clarity of implementation in the present Hadamard product approach. The main advantages of the present formulation are to separate the constant coefficient matrices G from nonlinear unknowns and provide an explicit unified matrix formulation-S. Thus, the FE formulations for general nonlinear problems are simplified greatly.

Except for a few very simple cases, the explicit direct formulation is not available in the conventional nonlinear FE analysis, namely $K\left(\vec{W}\right)$ in equation (2.9-30) is impossible to be explicitly expressed in algebraic form. In contrast, the present FE formulation-S has explicit matrix form and can be easily converted into the conventional FE formulation. For example, equation (2.9-30) can be restated

$$K_{n\times n}\vec{W} + \begin{Bmatrix} vec\left(G_1\right) \\ \vdots \\ vec\left(G_n\right) \end{Bmatrix}_{n\times 1} \left(\vec{W}\otimes\vec{W}\right) = \vec{F} \tag{2.9-31}$$

where vec( ) is the vector-function of a rectangular matrix formed by stacking the column of matrix into one long vector (Lancaster and Timenetsky). G_i ‘s are n×n symmetric matrices and can be obtained easily from the related rows of the matrix G in Eq. (2.9-28) through the invert process of vec( ) with n(n-1)/2 divisions. Furthermore, we have

$$K\left(\vec{W}\right)\vec{W} = \left(K_{n\times n} + \begin{Bmatrix} \vec{W}^T G_1 \\ \vdots \\ \vec{W}^T G_n \end{Bmatrix}_{n\times 1} \right)\vec{W} = \vec{F}\ . \tag{2.9-32}$$

Obviously, $K\left(\vec{W}\right)$ in the above equation is expressed as definite explicit matrix function.

It is also noted that Eqs. (2.9-11), (2.9-13), (2.9-25) and (2.9-28) have the same formulation-S form for the quadratic nonlinear problems. Therefore, the formulation-S is also believed to be in the most general sense the standard matrix formulation form in the ordinary and Kronecker product for the various nonlinear numerical computations. Moreover, due to obtaining identical formulation-S for a variety of nonlinear computation, the unified techniques can be well developed to solve the standard formulation-S.

In section 2.9.2, we will discuss the unified approaches to compute the Jacobi derivative matrix in the Newton-Raphson method for the solution of the formulation-S. In addition, unlike the foregoing FD, DQ, DC, pseudo-spectral and collocation methods, only the formulation-S is available in the FE, BE, spectral and Galerkin methods.

Considering heat conduction in a slab with a temperature dependent thermal conductivity as an example problem (Finlayson, 1980), the equation governing this system is stated as

$$\frac{d}{dx}\left[\left(1 + \alpha T\right)\frac{dT}{dx}\right] = 0 \qquad (2.9\text{-}33)$$

or simply

$$\frac{d^2T}{dx^2} + \alpha T\frac{d^2T}{dx^2} + \left(\frac{dT}{dx}\right)^2 = 0. \qquad (2.9\text{-}34)$$

with the boundary conditions

$$T(0) = T_0, \qquad T(L) = T_{L}. \qquad (2.9\text{-}35)$$

where α is a constant. The unknown T is first approximated by $\hat{T}$, namely,

$$\hat{T}^{(e)} = \sum_{j=1}^{m} N_j^{(e)}(x)T_j^{(e)} = \left\{N_j^{(e)}\right\}^T\left\{T_j^{(e)}\right\} = \vec{N}^{(e)T}\vec{T}^{(e)}. \qquad (2.9\text{-}36)$$

where $N_i(x)$ are the interpolation functions and T_i are the unknown nodal temperatures for an one-dimensional line element with m nodes. Applying the Galerkin's criterion, we have

$$\int_{x_1}^{x_m}\left[\frac{d^2\hat{T}^{(e)}}{dx^2} + \alpha\hat{T}^{(e)}\frac{d^2\hat{T}^{(e)}}{dx^2} + \alpha\left(\frac{d\hat{T}^{(e)}}{dx}\right)^2\right]N_j^{(e)}(x)dx = 0, \quad \text{i=1,2,....,N} \qquad (2.9\text{-}37)$$

where x_1 and x_m are the coordinates of the end nodes of the line element. Using integration by parts to the left-hand side of Eq. (2.9-37) and after some deductions, we have

$$\int_{x_1}^{x_m}\frac{dN_j^{(e)}}{dx}\frac{d\hat{T}^{(e)}}{dx} + \alpha\hat{T}^{(e)}\frac{dN_j^{(e)}}{dx}\frac{d\hat{T}^{(e)}}{dx}dx = N_j^{(e)}(x)\left(\frac{d\hat{T}^{(e)}}{dx} + \alpha\hat{T}^{(e)}\frac{d\hat{T}^{(e)}}{dx}\right)\Bigg|_{x_1}^{x_m} \qquad (2.9\text{-}38)$$

Substituting Eq. (2.9-36) into the above equation, we obtain

$$\int_{x_1}^{x_m}\frac{dN_j^{(e)}}{dx}\frac{\partial\vec{N}^{(e)T}}{\partial x}\vec{T}^{(e)} + \alpha\frac{dN_j^{(e)}}{dx}\left(\left[\vec{N}^{(e)T}\vec{T}^{(e)}\right]\circ\left[\frac{\partial\vec{N}^{(e)T}}{\partial x}\vec{T}^{(e)}\right]\right)dx = \int_{x_1}^{x_m}\frac{dN_j^{(e)}}{dx}\frac{\partial\vec{N}^{(e)T}}{\partial x}dx\vec{T}^{(e)}$$

$$+ \alpha\int_{x_1}^{x_m}\frac{dN_j^{(e)}}{dx}\left(\vec{N}^{(e)T}\otimes\frac{\partial\vec{N}^{(e)T}}{\partial x}\right)dx\left(\vec{T}^{(e)}\otimes\vec{T}^{(e)}\right) = N_mQ_m - N_1Q_1$$

$$(2.9\text{-}39)$$

in which $Q_1 = \frac{d\hat{T}^{(e)}}{dx} + \alpha\hat{T}^{(e)}\frac{d\hat{T}^{(e)}}{dx}$ at node 1 and $Q_m = \frac{d\hat{T}^{(e)}}{dx} + \alpha\hat{T}^{(e)}\frac{d\hat{T}^{(e)}}{dx}$ at node m are the natural boundary conditions, which are left unspecified. Equation (2.9-39) are the nonlinear formulation in the element and can be restated as

$$\underset{m\times m}{K^{(e)}}\vec{T}^{(e)} + \underset{m\times m^2}{G^{(e)}}\left(\vec{T}^{(e)}\otimes\vec{T}^{(e)}\right) = \left\{Q^{(e)}\right\}, \qquad (2.9\text{-}40)$$

which is similar to equation (2.9-27). $K^{(e)}$ and $G^{(e)}$ can be evaluated after the interpolation functions $N_i^{(e)}$'s are specified. $G_{ij} = \alpha\int_{x_1}^{x_m} \frac{dN_j^{(e)}}{dx} N_p^{(e)} \frac{\partial N_q^{(e)}}{\partial x} dx$, in which j=(p-1)m+q (p, q=1,2,...,m). The assembled FE equation for this case has the same form as Eq. (2.9-28), namely,

$$K_{n\times n}\vec{T} + G_{n\times n^2}\left(\vec{T}\otimes\vec{T}\right) = \vec{Q}\ . \tag{2.9-41}$$

where n is the total number of nodes in entire solution domain. Using the same procedure, more complex examples in multi-dimensional sense can be derived the similar FE formulation-S. For example, the static geometrically nonlinear equations for beam, plate and shell involve the cubic nonlinearity, the FE formulation can be generally stated in formulation-S form as

$$K_{n\times n}\vec{W} + G_{n\times n^3}\left(\vec{W}\otimes\vec{W}\otimes\vec{W}\right) = \vec{F}\ . \tag{2.9-42}$$

where $\vec{W}$ is the desired displacement vector. Obviously, equation (2.9-42) has the same form as the DQ formulation (2.9-17) for geometrically nonlinear beam.

Some remarks on the formulation-S in the FE method are given in the following:

a) The advantage of the formulation-S is that the known nonlinear coefficient matrix G are isolated from the unknowns vector and the formulation is obtained in an explicit matrix form. By using the formulation-S, the numerical integration in each iterative step is circumvented in the solution of the formulation-S. Thus, the computing effort is reduced greatly.

b) By using the Hadamard product and SJT product, the formulation-S in the finite element method can be derived not only from the method of weighted residuals but from variational principles and energy balances approach.

c) The finite element equation for nonlinear time-dependent problems can be derived by using the Hadamard product in the same way. The resulting FE formulation is ordinary differential equations in time and has the formulation-S form, in which the time-derivative vector is included.

d) The formulation-S is given in global sense, which is different from the traditional incremental FE method for nonlinear problems (Bath, 1982; Yin, 1987). However, when the external force is gradually imposed by a small value in each incremental step, the present FE formulation-S will be equivalent to the incremental FE equations, namely, the Total-Lagrangian (T. L.) method if the node coordinates are invariant in each loading step or Updated-Lagrangian (U.L.) method if the node coordinates must be refreshed in each step. In addition, the matrix G in formulation-S is always unvarying in the T.L. method and must be recalculated in each step in the U.L. method.

e) The conventional nonlinear FE method may be too complex mathematically for routine applications (Bath, 1982; Yin, 1987) and thus are not easily learned or used. In contrast, the present Hadamard product approach is an explicit and simple matrix analysis technique and thus the complexity in the conventional method is circumvented. The formulation-H can be also derived from the finite element methods based on the Galerkin collocation, pseudo-spectral (spectral element), differential quadrature (Striz et al., 1994), orthogonal collocation (Finlayson, 1980), etc.

We here point out a incorrect conclusion in Noguchi and Hisada (1995) on the T.L and U.L methods. By using tensorial components in the natural coordinate system, Noguchi and Hisada (1995) proposed an unified FE formulation in T.L. and U.L. methods to handle geometrically nonlinear problems. The basis idea in this paper is that "the tangent stiffness matrices and the numerical solutions obtained by these methods should be identical in geometrically nonlinear problems" . However, as was pointed out by Lü et al. (1995), the T.L and U.L. methods use formally the same curvature strain formula, but in fact, the T.L. method considers only the linear terms in the curvature strain, while the curvature strain used in the U.L. method includes the nonlinear terms implicitly due to the coordinate transformation (Lü et al., 1995; Cook, 1974). Mattiasson et al. (1986) also pointed out the limitations of the T.L. method. Several geometric nonlinear problems provided by Lü et al. (1995) and Mattiasson et al. (1986) also showed the obviously different results obtained by these two methods under situations of larger deformation.

2.9.2. Jacobian Derivative Matrix of the Formulation-S

In this section we give explicit direct formulas for computing the analytical Jacobian derivative matrix in the solution of the quadratic and cubic nonlinear formulation-S and applied to numerical example successfully.

The formulation-S for the quadratic nonlinear differential problems was converted into equation (2.9-32) in section 2.9.1. The equation can be further restated as

$$\varphi(\vec{W}) = K_{n\times n}\vec{W} + \begin{Bmatrix} \vec{W}^T G_1 \vec{W} \\ \vdots \\ \vec{W}^T G_n \vec{W} \end{Bmatrix}_{n\times 1} - \vec{F} = 0 \qquad (2.9\text{-}43)$$

where $\vec{W}^T$ is the transpose of vector $\vec{W}$. G_i‘s are n×n symmetric matrices. According to the rule in differentiation of matrix function (Ni, 1984), the Jacobian derivative matrix for the above equation can be obtained by the following direct computing formula:

$$\frac{\partial\varphi\{\vec{W}\}}{\partial\vec{W}} = K_{n\times n} + 2\begin{Bmatrix} \vec{W}^T G_1 \\ \vdots \\ \vec{W}^T G_n \end{Bmatrix}_{n\times 1} \qquad (2.9\text{-}44)$$

where $\frac{\partial}{\partial\vec{W}}$ is Jacobi derivative matrix operator in the Newton-Raphson method. The computational effort using the above formula is only $n^2(n+1)/2$ multiplications. We should notice that the formula (2.9-44) is very easy to be computed parallelly for each row. The further reduction of computational effort is still possible by using the properties of matrices G_i’s such as sparseness.

Considering the formulation-S for the cubic nonlinear problems, for example, equation (2.9-14) in 2.9.1

$$L_{n\times n}\vec{U} + R_{n\times n^3}\left(\vec{U}\otimes\vec{U}\otimes\vec{U}\right) + \vec{F} = 0 \qquad (2.9\text{-}45)$$

can be restated as

$$\phi\left(\vec{U}\right) = L_{n\times n}\vec{U} + \begin{Bmatrix} \vec{U}^T R^1_{n\times n^2}\left(\vec{U}\otimes\vec{U}\right) \\ \vdots \\ \vec{U}^T R^n_{n\times n^2}\left(\vec{U}\otimes\vec{U}\right) \end{Bmatrix}_{n\times 1} + \vec{F} = 0 \qquad (2.9\text{-}46)$$

Similarly, the Jacobian matrix of the above equation can be evaluated by

$$\frac{\partial\phi\{\vec{U}\}}{\partial\vec{U}} = L_{n\times n} + \begin{Bmatrix} \dfrac{\partial R^{1}_{n\times n^2}\left(\vec{U} \otimes \vec{U}\right)}{\partial\vec{U}} \\ \vdots \\ \dfrac{\partial\vec{U}R^{n}_{n\times n^2}\left(\vec{U} \otimes \vec{U}\right)}{\partial\vec{U}} \end{Bmatrix} \qquad (2.9\text{-}47)$$

in which

$$\frac{\partial \vec{U}^T R^i_{n\times n^2}\left(\vec{U}\otimes\vec{U}\right)}{\partial \vec{U}} = \left[R^i_{n\times n^2}\left(\vec{U}\otimes\vec{U}\right)\right]^T + \vec{U}^T \frac{\partial R^i_{n\times n^2}\left(\vec{U}\otimes\vec{U}\right)}{\partial \vec{U}}.$$

Furthermore, we have

$$\frac{\partial \phi\left(\vec{U}\right)}{\partial \vec{U}} = L + 3\begin{bmatrix} \vec{U}^T R_{11}\vec{U} & \vec{U}^T R_{12}\vec{U} & \cdots & \vec{U}^T R_{1n}\vec{U} \\ \vec{U}^T R_{21}\vec{U} & \vec{U}^T R_{22}\vec{U} & \cdots & \vec{U}^T R_{nn}\vec{U} \\ \vdots & \vdots & \vdots & \vdots \\ \vec{U}^T R_{n1}\vec{U} & \vec{U}^T R_{n2}\vec{U} & \cdots & \vec{U}^T R_{nn}\vec{U} \end{bmatrix} \tag{2.9-48}$$

Similar formulas for higher order nonlinear problems can be easily obtained.

Consider the equation of the quadratic nonlinearity (Chen and Zhong, 1996c; Wright, 1964)

$$y_{,x}^2 + y_{,xx} + yy_{,xx} = 0, \qquad y(0)=0, y(1)=1, \tag{2.9-49}$$

Using the DQ method, we have

$$\vec{y}\,' = \overline{A}\vec{y} + \vec{a}. \qquad \vec{y}\,'' = \overline{B}\vec{y} + \vec{b}. \tag{2.9-50}$$

where $\overline{A}$ and $\overline{B}$ are the weighting coefficient matrices, modified by the boundary conditions, for the first and second derivatives in the DQ method, respectively. $\vec{a}$ and $\vec{b}$ are the constant vector dependent on the boundary conditions. We have the formulation-S for this case:

$$L_{n\times n}\vec{y} + Q_{n\times n^2}\left(\vec{y} \otimes \vec{y}\right) + \vec{F} = 0 \tag{2.9-51}$$

where

$$L_{n\times n} = I\,b + 2\overline{A}\Diamond\vec{a} + \overline{B}$$

$$Q_{n\times n^2} = E_n^T\left(I \otimes \overline{B} + \overline{A} \otimes \overline{A}\right). \tag{2.9-52}$$

$$\vec{F} = 1 + \vec{a}^{\circ 2} + \vec{b}$$

I is the unit matrix. The solutions for the linear differential equation

$$y_{,xx} = 0; \qquad y(0) = 0, \; y(1) = 1 \tag{2.9-53}$$

is used as the initial guess of the Newton-Raphson method. The Jacobian matrix is computed by using equation (2.9-44). Six Chebyshev grid points in the DQ method are used for this case and the convergence is achieved with four iterations. The relative error of the numerical results obtained are no more than 0.001.

We also recalculate the same cubic nonlinear equation (2.5-1a, b) in section 2.5 using formulas (2.9-47) successfully, and the numerical results are coincident with the ones in section 2.5. The numerical examples demonstrate the efficiency and simplicity of the present formulas computing the Jacobian matrix for the standard formulation-S. More work on the nonlinear formulation-S may prove to be beneficial.

However, as for the formulation-H, the SJT product may be a more effective and simple approach for the evaluation of the Jacobian matrix. It is obvious that the operation rules on the SJT product are applicable for the pseudo-spectral and collocation methods. It is shown in section 2.3 that the SJT product can compute accurately the Jacobian matrix in chain rule similar to those in differentiation of scalar function with minimum computing effort among all existing approaches. Unfortunately, however, the SJT product seems to be not amenable to the evaluation of the Jacobian matrix of the more general formulation-S. Traditionally, the Jacobian matrix is often obtained by numerical derivative computations and thus the speed of convergence in the Newton-Raphson method is affected. In contrast, the SJT product approach and the presented direction formulas give the analytical Jacobian matrix for the formulation-S and formulation-H. It is also noted that the formulation-H is more easily obtained for the nonlinear computations. Therefore, the formulation-H is preferred for the nonlinear computations of the FD, DQ, DC, pseudo-spectral, collocation methods and some variants.

2.9.3. Perturbed Error Analysis

The condition number of coefficient matrix can indicate the effect of round-off errors on the accuracy of the solution of linear simultaneous algebraic equations, and is closely related to stability analysis of numerical computation (Wilkinson, 1964; Burden et al., 1970; Shi, 1980). In this section we will discuss the relation between the condition number and the perturbed error bound in the solution of the nonlinear formulation-H and -S.

For the quadratic nonlinear problems, the presented formulation-S in the ordinary and Kronecker product is

$$K_{n\times n}\vec{U} + G_{n\times n^2}\left(\vec{U} \otimes \vec{U}\right) = \vec{F}. \tag{2.9-54}$$

We will consider the effects of perturbing the right side vector $\vec{F}$. Suppose first the above equation is perturbed so that matrices K and G are kept fixed and δF is added to $\vec{F}$. Thus,

$$K\left(\vec{U} + \delta\vec{U}\right) + G\left(\left(\vec{U} + \delta\vec{U}\right) \otimes \left(\vec{U} + \delta\vec{U}\right)\right) = \vec{F} + \delta\vec{F}, \tag{2.9-55}$$

then

$$\delta\vec{U} = K^{-1}\delta\vec{F} - K^{-1}G\left(\vec{U}\otimes\delta\vec{U} + \delta\vec{U}\otimes\vec{U} + \delta\vec{U}\otimes\delta\vec{U}\right) \tag{2.9-56}$$

$\delta\vec{U}\otimes\delta\vec{U}$ in the above equation is relatively fairly small in comparison to other terms in the above equation and thus neglected in the following analysis, and

$$\left\|\vec{U}\otimes\delta\vec{U}\right\| = \left\|\vec{U}\right\|\left\|\delta\vec{U}\right\| \tag{2.9-57}$$

We have

$$\left\|\delta\vec{U}\right\| \le \frac{\left\|K^{-1}\right\|\left\|\delta\vec{F}\right\|}{1 - 2\left\|K^{-1}\right\|\left\|G\right\|\left\|\vec{U}\right\|} \tag{2.9-58}$$

Form Eq. (2.9-54),

$$\left\|\delta\vec{F}\right\| \le \left(\left\|K\right\| + \left\|G\right\|\left\|\vec{U}\right\|\right)\left\|\vec{U}\right\|$$

and

$$\frac{1}{\left\|\vec{U}\right\|} \le \frac{\left\|K\right\| + \left\|G\right\|\left\|\vec{U}\right\|}{\left\|\delta\vec{F}\right\|} \tag{2.9-59}$$

Multiplying Eqs. (2.9-58) and (2.9-59) provides

$$\frac{\left\|\delta\vec{U}\right\|}{\left\|\vec{U}\right\|} \le \frac{\left\|K^{-1}\right\|\left\|K\right\| + \left\|K^{-1}\right\|\left\|G\right\|\left\|\vec{U}\right\|}{1 - 2\left\|K^{-1}\right\|\left\|G\right\|\left\|\vec{U}\right\|}\frac{\left\|\delta\vec{F}\right\|}{\left\|\vec{F}\right\|} = \frac{\kappa_L(K) + \kappa_N(K,\ G)\left\|\vec{U}\right\|}{1 - 2\kappa_N(K,\ G)\left\|\vec{U}\right\|}\frac{\left\|\delta\vec{F}\right\|}{\left\|\vec{F}\right\|} \tag{2.9-60}$$

where $\kappa_L(K)$ is the conventional condition number of linear system, $\kappa_N(K,\ G)$ is defined as the nonlinear condition number. The quantity on the left of Eq. (2.9-60) may be considered a measure of the relative disturbance of $\vec{U}$. The inequality provides a bound on this relative disturbance in terms of the relative disturbance of $\vec{F}$. Removing the nonlinear term in Eq. (2.9-54), we can obtain the traditional linear formula:

$$\frac{\left\|\delta\vec{U}\right\|}{\left\|\vec{U}\right\|} \le \left\|K^{-1}\right\|\left\|K\right\|\frac{\left\|\delta\vec{F}\right\|}{\left\|\vec{F}\right\|} = \kappa_L(K)\frac{\left\|\delta\vec{F}\right\|}{\left\|\vec{F}\right\|} \tag{2.9-61}$$

Compared with Eq. (2.9-61), the relative disturbance bound in the formula (2.9-60) depends on not only $\kappa(K)$ but also $\kappa_N(K,\ G)$ and the norm of unknown vector x. Thus, the analysis for nonlinear problems are much more complex than that for linear problems. Let the nonlinear condition number $\kappa_N(K,\ G)$ zero, Eq. (2.9-60) is equivalent to Eq. (2.9-61).

The following analyzes the nonlinear formulation-H.

Theorem 2.9.3.1. If matrices A=[a_{ij}] and B=[b_{ij}]∈$C^{N\times M}$, then

$$\|A \circ B\|_2 \le \|A\|_2\|B\|_2 \tag{2.9-62}$$

where $\|\ \|_2$ denotes the spectral norms. This is famous Schur theorem on the Hadamard product. Note that the matrices A and B involved in numerical computation are in general N×1 dimensional vector, i.e., A, B∈$C^{N\times 1}$. So it is straightforward that the inequality (2.9-62) exists for the other norms in the present study, and thus the particular choice of norm is immaterial, we will omit the subscript 2 in the latter discussion.

For example,

$$K\vec{U} + \left(A\vec{U}\right) \circ \left(B\vec{U}\right) = \vec{F} \tag{2.9-63}$$

is perturbed. A and B are kept fixed, and F is added to δF, namely,

$$K\left(\vec{U}+\delta\vec{U}\right)+\left(A\left(\vec{U}+\delta\vec{U}\right)\right)\circ\left(B\left(\vec{U}+\delta\vec{U}\right)\right)=\vec{F}+\delta\vec{F} \tag{2.9-64}$$

Thus,

$$\delta\vec{U} = K^{-1}\delta\vec{F} - K^{-1}\left[\left(A\vec{U}\right)\circ\left(B\delta\vec{U}\right)-\left(A\delta\vec{U}\right)\circ\left(B\vec{U}\right)-\left(A\delta\vec{U}\right)\circ\left(B\delta\vec{U}\right)\right] \tag{2.9-65}$$

Multiplying Eqs. (2.9-58) and (2.9-64) and applying theorem 2.9.3.1, we have

$$\frac{\|\delta\vec{U}\|}{\|\vec{U}\|} \le \frac{\|K^{-1}\|\|K\| + \|K^{-1}\|\|A\|\|B\|\|\vec{U}\|}{1 - 2\|K^{-1}\|\|A\|\|B\|\|\vec{U}\|}\frac{\|\delta\vec{F}\|}{\|\vec{F}\|} = \frac{\kappa_L + \kappa_N\|\vec{U}\|}{1 - 2\kappa_N\|\vec{U}\|}\frac{\|\delta\vec{F}\|}{\|\vec{F}\|}. \tag{2.9-66}$$

$\left(A\delta\vec{U}\right)\circ\left(B\delta\vec{U}\right)$ is relatively fairly small value in comparison to other terms in Eq. (2.9-65) and thus neglected in the analysis. The nonlinear condition number here is $\kappa_N(K,A,B)$ similar to $\kappa_N(K,\ G)$ in Eq. (2.9-60).

Equations (2.9-60) and (2.9-66) may establish the basis for stability and error analysis in nonlinear numerical computations. The further applications are under active study.

2.10. Concluding Remarks

In the closing section we will discuss the potential advantages of the DQ and DC methods for the nonlinear problems. After that, some remarks on the Hadamard product and SJT product techniques are given.

2.10.1. Some advantages in applying the DQ and DC methods for nonlinear problems

Some researchers (Malik and Civan, 1994; Wang, 1995) have pointed out that the DQ and DC methods may be especially attractive for nonlinear computations. But they did not provide some theoretical analysis to support their arguments. We here discuss some inherent merits in the DQ and DC methods for nonlinear computations.

The DQ and DC methods belong to the global numerical techniques. It is well known that the most distinctive feature of the nonlinear problems is their global correlation in comparison to the linear problems. In other words, the function values far from each other in entire computational domain may have much larger relative affection for nonlinear problems. Of course, for linear problems, there exists similar each other dependence, but that is weaker. The finite element and finite difference methods are derived from local interpolation technique. The global methods use much more function values to approximate the function value at certain grid point, while the FE and FD methods only use less function information, in which the solution at a grid point is approximated only by using those dependent-variable values at adjacent points. The main shortcomings of the global methods are their computational stability and the applicability for problems with complex geometries. The former can be circumvented by using the proper basis functions for problems of interest, the latter can be overcome to somewhat extent by coordinate mappings or multidomain approaches. If the solution for problems of interest is sufficiently smooth and we choose proper basis functions, the global methods produce more reliable results and has more rapid rate of convergence than the FE and FD methods, especially for nonlinear problems. From the resulting formulation standpoint, the FE and FD methods result in a sparsely banded coefficient matrix, while the DQ method generates algebraic equations with full characteristic matrices in one dimensional problems. But for two-dimensional domain problems, Civan and Sliepcevich (1984b) pointed out that the Jacobian matrix in the DQ algebraic formulation contained many null elements, nearly a half of all elements but much less than in the FD and FE formulations. Unlike the DQ method, the DC method is a really global numerical method for multi-dimensional problems. The DQ method is a special case of the DC method in one dimension. The resulting coefficient matrices in the DC formulation for multi-dimensional problems contain much less null element. Therefore, the DC method may be the most efficient numerical technique for multidimensional nonlinear problems. Some applications also proved much higher efficiency of the DQ and DC methods for the nonlinear problems than the conventional FE and FD methods. In addition, it should be pointed out that the DQ and DC methods are also generally more effective for the linear problems than the FE and FD methods. The DC method may be especially efficient for differential equations whose derivatives with respect to one coordinate is not only dependent on the function value at that coordinate direction.

On the other hand, the other global numerical methods (Rayleigh-Ritz, Galerkin, etc.) also have similar inherent efficiency for nonlinear computations. However, these methods require one to select initial trial functions satisfying boundary conditions for problems considered, which is not an easy task for many problems in practice. In addition, these methods need more strenuous formulation effort. In contrast, the DQ and DC method can easily and exactly satisfy a variety of boundary conditions and require much less formulation effort. Recently, the spectral and pseudo-spectral methods have been extensively used in practical engineering especially for the numerical solutions of fluid dynamic problems. The spectral and pseudo-spectral methods are also belong to global numerical methods and are efficient for many linear and nonlinear problems. Quan and Chang (1989a) have pointed out that the DQ-type methods are basically equivalent to the collocation and pseudo-spectral methods. But the DQ and DC methods directly compute the desired function values at grid points rather than spectral variables. Thus, they are more explicit and simple for some practical applications. Moreover, the pseudo-spectral methods require heavy and complicated formulation effort and lack the ease of implementation of the DQ and DC methods (Mansell et al., 1993). Also, as was discussed earlier, the desired spectral coefficients in these methods usually have no physical significance. Therefore, assumed initial values in the iterative solution of the nonlinear problems are inherently poor guess, and the computational effort is aggravated.

The proper basis functions can improve the convergence rate and reliability of the DQ method (Chang et al., 1993; Striz et al, 1995), but it should be emphasized that the choice of the basis functions in the DQ and DC methods is different from the Rayleigh-Ritz and Galerkin methods. The DQ and DC methods need not consider the boundary conditions of certain problems to choose the basis functions. Thus, the effort to determine the basis functions is very little. Generally, we can categorize the problems into several sorts. For example, the harmonic functions are chosen as the basis functions for periodic problems, the family of Bessel functions for buckling, deflection and vibration of circular and annular plates. The DQ and DC methods using the polynomial basic functions are usually effective and reliable for a variety of problems if the zeros of Chebyshev, Gauss or Legendre polynomials are adopted as the grid spacing. The idea using split range polynomial expansions as the basic functions proposed by Mansell et al. (1993) is applicable to the DQ and DC methods for dealing with problems involving steep gradients and discontinuities. For the definite basis functions, the DQ and DC weighting coefficients for certain grid spacing need be computed only once and are independent of any special problems. Therefore, these weighting coefficients can be used repeatedly for various problems.

Based on the foregoing discussions, we can conclude that the DQ and DC methods has the superb accuracy, efficiency, convenience and the great potential for nonlinear numerical computations.

2.10.2. Some remarks on the Hadamard product and SJT product

The conventional nonlinear algorithms have been rather adequately studied with the framework of linear algebra and relative matrix approaches. As was pointed out earlier, the approaches do not work very well for this task. In this chapter we hope to present a new framework for nonlinear computations of the DQ-type methods. The principal contributions of this chapter are to introduce the Hadamard product of matrices to the nonlinear computations of the DQ method as well as other numerical techniques and to first present new SJT product to compute the Jacobian derivative matrix in the Newton-Raphson method. The specialist in numerical computation field are seldom familiar with the Hadamard product. As was pointed out by Horn (1990), the Hadamard product was even rarely involved in the standard matrix and linear algebraic text book. The objective of this chapter is to recognize the usefulness and generality of the Hadamard product in nonlinear computations. Using the Hadamard product, the nonlinear differential equations are easily formulated by various numerical methods. Moreover, regardless of the method we choose to compute the nonlinear problems, the resulting formulation can have the same formulation-S form. The formulation-S isolates the known coefficient matrix and unknown nonlinear vectors and has an explicit matrix form. Therefore, the solution and analysis of the formulation-S are much easier than that of the conventional nonlinear formulations. For the finite difference, differential quadrature, differential cubature, pseudo-spectral and collocation methods as well as the finite element methods based on these numerical techniques, we can obtain either the formulation-S or the formulation-H. The formulation-H is preferred for these methods. The most important point in this study is that a computational attractiveness is shown to develop the unified techniques for the analysis and computation of the standard formulation-H and formulation-S.

The approaches are developed to efficiently compute the Jacobian matrix and analyze the perturbed error of the formulation-H and -S. The simple iteration method become a competitive alternative to the Newton-Raphson method due to the application of the Hadamard power and function. The SJT product may be very significant because the Jacobian matrix is often used in many engineering areas. The decoupling computations by means of the Hadamard and SJT product are of extreme importance in the solution of large nonlinear systems. The Hadamard product, power and function are shown to be innovative and powerful concepts in nonlinear numerical computations. The most important feature of the Hadamard product may be its ability to express the nonlinear relation in the problems of interest. But in fact it seems to be seldom applied in practical engineering. Though we restrict attention here to numerical computations of nonlinear differential equations, the work will possibly bring impulse for some new development in the Hadamard product and its applications in various science and engineering areas, in which nonlinear analysis can not be circumvented. The Hadamard product may be of vital importance in ***"nonlinear algebra***" rather than in the conventional linear algebra.

The practical applications of the DQ method to nonlinear problems are still few in comparison to the other numerical methods. The present work may provide a guidance to more studies in this field. In section 4.5 we will show the practical applications of the techniques in the DQ solution of geometrically nonlinear bending of orthotropic plates.

CHAPTER 3

TRUNCATION ERROR AND GRID SPACINGS

3.1. Introduction

As was pointed out by Bert and Malik (1996d), the studies in the error aspects of the DQ method have been neglected. The work has been not reported so far except Bellman et al. (1972), Jang et al. (1989), Chen et al. (1994, 1996b) and He and Wang (1995). Bellman et al. (1972) presented truncation error formulas based on Rolle's theorem. However, the formulas do not involve the practical grid interval and are too inaccurate for practical use. Therefore, they are not significant for many practical purpose. Based on Bellman's formulas, Jang et al. (1989) briefly discussed the error estimates for the DQ analysis of a beam. Chen et al. (1994, 1996b) derived a new explicit truncation error formulas for the first order derivative by using interpolation technique and discuss in general the truncation error distribution problems. Then, by using the same technique, He and Wang (1995) compared maximum truncation errors of the 1st order derivative in using several types of sampling points to find the optimal grid spacings. Recently, Chen, Zhong and He (1996d) gave the simple and explicit truncation error formulas for the DQ approximation of various order derivatives. It is also noted that the choice of the sampling points is deeply related to the truncation error. It is known that the optimal grid spacing in the polynomial approximate if a function is to satisfy the mimimax principle, i.e., the absolute truncation errors at various grids are basically uniform (Atkinson, 1978). The optimal approximation of derivatives of a function is an extremely important problem. However, research in this field is still fairly poor so far. In this study, we discuss the truncation error and sampling point problems in the DQ method through trial and error. Seven general rules for choosing sampling points are presented. The relations between the optimal types of grid spacings and truncation errors are discussed. Since the DQ method is in fact equivalent to the collocation and pseudo-spectral methods, the truncation error formulas and the rules for selecting sampling points are also applicable for these two methods. It is worth pointing out that the proposed truncation error formulas are also important for analyzing the convergence speed of the DQ method.

The zeros of orthogonal polynomials such as the Legendre or Chebyshev polynomials do not include the end points of the normalized domains, and, thus, the end points have to be forced if these points are needed for the boundary conditions. The DQ method using such grid points is equivalent to the so-called orthogonal collocation methods. Bellman et al. (1975) applied the spline technique to avoid this boundary point difficulty. Bert and Malik (1996d) suggested that the Chebyshev-Gauss-Lobatto points be a better choice in that the grid spacing automatically includes the end points. However, as was pointed out by Quan and Chang (1989a), the zeros of the Chebyshev or Legender polynomials can be used to reach the end points by using a simple linear algebraic transformation, and direct computing formulas for weighting coefficients of such grid points are also presented. Their study also showed that the DQ method using all zeros of the Chebyshev polynomials without additional forced ends is more efficient than using the modified Chebyshev grid points. However, the formulas given by Quan and Chang (1989a) is still somewhat complex. In the present study, the simplified formulas for computing weighting coefficients of such grid spacing as well as other often used grid spacings are provided.

3.2. Truncation Error in the DQ Method

Based on the fact that the DQ method is a polynomial approach (Quan and Chang, 1989a, b), new truncation error formulas are derived and their distribution in the entire variable domain are discussed in this section.

Bellman et al. (1972) presented a truncation error formula for the DQ approximation to the second derivative:

$$\left|R^{(2)}(x)\right| \le \frac{K}{(N-2)!}\prod_{j=1}^{N-2}\left(x-\bar{x}_j\right) \le \frac{Kh^{N-2}}{(N-2)!} \tag{3.2-1}$$

where K is the maximum value of the Nth derivative of the function f(x), $\left|f^{(N)}(x)\right| \le K$. $x_j \angle \bar{x}_j \angle x_{j+2,}$ x_j 's are the grid coordinate. h is the maximum interval on the grid, h=max$\left|x-\bar{x}_j\right|$. The formula was derived by using Rolle's theorem. There exist similar truncation error formulas for the 1st and other higher order derivatives. The major drawback of the above formula is that relation between the truncation error and the grid interval has not be clearly exposed, since $\bar{x}_j$'s are not definite value as the grid coordinate x_j. Therefore, h in the formulas is too vague for many practical purpose.

Quan and Chang (1989a, b), Shu and Richards (1992) and Bert, Wang and Striz (1993) derived the DQ method from the Lagrangian interpolation formulas. Following this idea, Chen (1994) and Chen, Yu, Wang (1996b) proposed a different truncation error formulas in the DQ method for the 1st order derivative.

If a function can be approximated by a Lagrangian polynomial, we have

$$f(x) = \sum_{j=1}^{N} p_j(x) f(x_j) + R(x), \qquad j = 1, 2, \ldots, N. \tag{3.2-2}$$

where $p_j(x)$'s are Lagrangian interpolation basis polynomials, R(x) is the truncation error, namely,

$$R(x) = \frac{f^{(N)}(\zeta)W(x)}{N!} \tag{3.2-3}$$

in which $W(x) = \prod_{i=1}^{N}(x - x_i)$. Hence,

$$f'(x_i) = \sum_{j=1}^{N} p_j'(x_i) f(x_j) + R' = \sum_{j=1}^{N} A_{ij} f(x_j) + R'(x_i), \tag{3.2-4}$$

where A_{ij}'s are the DQ weighting coefficients of the 1st order derivative, x_i's are discrete grid point coordinates, $R'(x_i)$ is the truncation error of the DQ method to approximate the 1st order derivative. Thus, we have

$$R'(x_i) = \frac{f^{(N)}(\xi)W'(x_i)}{N!}, \qquad \text{i=1,2,...,N.} \tag{3.2-5}$$

where $W'(x)$ depends on the grid spacings. It can be observed that the truncation error formula (3.2-5) is different from one given by Bellman et al. (1972).

In the subsequent study we will derive new formulas of the truncation error for high order derivatives. A_{ii}, B_{ij} and C_{ij} herein represent the DQ weighting coefficients for the 1st, 2nd and 3rd order derivatives, respectively. Differentiate equation (3.2-2) to the second order and let $x=x_i$, we have

$$\left|R''(x_i)\right| \le K\left|2W'(x_i) + W''(x_i)\right| \frac{1}{N!} \qquad \text{i=1,2,...,N.} \tag{3.2-6}$$

where K=max$\left(\left|f^{(N)}(x)\right|, \left|f^{(N+1)}(x)\right|\right)$, R'' (x_i) is the corresponding truncation error in the DQ method. According to equation (8b) in Shu and Richards (1992), namely

$$A_{i\,i} = \frac{W''(x_i)}{2W'(x_i)} \tag{3.2-7}$$

where A$_{ii}$'s are the diagonal entries in the DQ weighting coefficient matrix for the 1st order derivative. We have

$$\left|R''(x_i)\right| \;\le\; 2K\left|1 \;+\; A_{i\,i}\right| \frac{\left|W'(x_i)\right|}{N\,!}, \tag{3.2-8}$$

According to

$$B_{i\,i} \;=\; \frac{W'''(x)}{3W'(x)} \quad \text{and} \quad C_{i\,i} \;=\; \frac{W^{(4)}(x)}{4W'(x)}. \tag{3.2-9}$$

By using the similar procedure, we have

$$\left|R'''(x_i)\right| \le 3K_3\left|1+2A_{ii}+B_{ii}\right|\frac{\left|W'(x_i)\right|}{N!} \tag{3.2-10}$$

$$\left|R^{(4)}(x_i)\right| \le 4K_4\left|1+3A_{ii}+3B_{ii}+C_{ii}\right|\frac{\left|W'(x_i)\right|}{N!} \tag{3.2-11}$$

where K$_4$=max$\left(\left|f^{(N)}(x)\right|,\left|f^{(N+1)}(x)\right|,\left|f^{(N+2)}(x)\right|,\left|f^{(N+3)}(x)\right|\right)$, K$_3$=max$\left(\left|f^{(N)}(x)\right|,\ \left|f^{(N+1)}(x)\right|,\right.$ $\left.\left|f^{(N+2)}(x)\right|\right)$.

In this section, we only discuss the zeros of the Chebyshev polynomials and the equally spaced grid points. For example, consider truncation error of the 2nd order derivative under the equally spaced grid points, we have

$$\left|R''(x_i)\right| \;\le\; 2K\left|1 \;+\; A_{i\,i}\right| \frac{\tau^{N-1}}{NC_{N-1}^{i-1}} \tag{3.2-12}$$

where $\tau \;=\; \dfrac{b \;-\; a}{N \;-\; 1}$ is the uniformly grid interval. b and a are the upper and below limits of variable x, respectively, namely x∈[a, b], C_{N-1}^{i-1} denotes combination computation. Note that τ is accurate value greatly different from imprecise h in equation (3.2-1). Therefore, the present formula indicates more explicitly the truncation errors of the approximate solutions at various sampling points in terms of the grid interval. For the grid spacing using the zeros of the Chebyshev polynomials of the first kind, we have

$$\left|R''(x_i)\right| \le \frac{2K\left|1+A_{ii}\right|}{2^{N-1}(N-1)!\sqrt{1-r_i^2}}\left(\frac{b-a}{d}\right)^{N-1} \tag{3.2-13}$$

where $d = 2\cos\dfrac{\pi}{2N}$ is the distance between the last and first roots of the N order Chebyshev polynomial in the normalized range (-1, 1), r$_i$ is the corresponding ith root of the Chebyshev polynomial, x∈[a, b]. The convergence speed using this grid spacing is faster than using the equally spaced grids (Quan and Chang, 1989a, b; Bert, Wang and Striz, 1993).

To facilitate comparison with the equally spaced grid spacing, the truncation error formula (3.2-13) for the Chebyshev grid spacings is restated as

$$\left|R''(x_i)\right| \le \frac{K(1+A_{ii})N^{N-1}}{(2d)^{N-1}(N-1)!\sqrt{1-r_i^2}}\left(\frac{b-a}{N}\right)^{N-1} = \frac{K(1+A_{ii})N^{N-1}}{(2d)^{N-1}(N-1)!\sqrt{1-r_i^2}}\,\tau^{N-1} \tag{3.2-14}$$

where τ=(b-a)/N. The general truncation error formulas for the 1st, 2nd, 3rd and 4th order under various grid spacings can be expressed as

$$\left|R_i^{(m)}\right| \le K_{(m)} err\,\tau^{N-1}, \quad \text{m=1, 2, 3, 4} \qquad \text{i=1,2,....,N} \tag{3.2-15}$$

in which err denotes the corresponding error constants.

Considering equally spaced grid points, we have

$$\left|R'(x_1)\right| \le \frac{K}{N(N-1)^{N-1}} \quad \text{and} \quad \left|R'(x_N)\right| \le \frac{K}{N(N-1)^{N-1}} \tag{3.2-16}$$

for both ends and

$$\left|R'\left(x_{N/2}\right)\right| \le \frac{(N/2-1)!\,(N/2)!\;K}{N\,!\,(N-1)^{N-1}} \tag{3.2-17}$$

for the center points if N is an even number.

Comparing the truncation error formulae (3.2-16) for the ends with (3.2-17) for the center point, it may be concluded that largest truncation error could possibly occur in the vicinity of the ends if equally spaced grid points are used in the DQ method. The two examples, which were Eqs. (4) and (5) provided by Civan and Sliepcevich (1986) as accuracy tests of the DQ method, can be as the numerical demonstration for the present error analysis (see fig. 1 and fig. 2 of Civan and Sliepcevich (1986)). Also a similar error distribution is expected for unequally spaced grids such as defined by the zeros of the shifted Chebyshev or Legendre polynomials.

The truncation error constants at all grid points for the 2nd order derivative under N=10 are given in table 3-1 for the equally spaced and the Chebyshev grids, where i denotes the sequence of the grid points. It can be found that the error constants except for those at end points for the Chebyshev grids are in general more smooth than those for the equally spaced grids. This compares favorably with the fact that the DQ method using the Chebyshev grid points can produce more accurate results than using the equally spaced grid points. It is fact that maximum diagonal entries in weighting coefficient matrices A_{ij}, B_{ij} and C_{ij} come upon at the end points. So the largest error constant may appear at the end points according to Eqs. (3.2-11), (3.2-16) and (3.2-17). In table 3-1 we can note that the largest error constants indeed locate at the ends of these two grid spacings. Therefore, it can be concluded that the largest truncation error may much probably occurs in the vicinity of the ends. Also, it is known that the Chebyshev grid spacing is very efficient for some problems among all existing grid spacings (Quan and Chang, 1989b) but not optimal for the DQ method as well as the pseudo-spectral and collocation methods since the error constants at the vicinity of the ends are obviously larger than at other grid points. Similar to min(max(W(x)) principle in the polynomial approximation of a function by using the Chebyshev grid points, it is very significant work to find the grid spacing which satisfies min(max $W'(x_i)$) at inner grid points. The further discussions on this issue can be found in section 3.4.

Table 3-2 displays the error constants at the end point N for the equally spaced and the Chebyshev grid points under N=5, 10, 15, 20. err(1), err(2), err(3) and err(4) denote the corresponding truncation error constants for the 1st, 2nd, and 3rd and 4th order derivatives, respectively. Note that the truncation errors for various order derivatives are all O($\Delta\tau^{N-1}$) in the DQ method but have greatly different the error constants. As can be seen from table 3-2, the error constants for high order derivative are in general larger. As the number of grid points increases, the error constants for the second order derivative increase in the equally spaced grid points but decrease in the Chebyshev grid points. So, if using the larger number of grid points, the Chebyshev grid spacing is preferred. In addition, it is also observed that the diagonal elements in weighting coefficient matrices A_{ij}, B_{ij} and C_{ij} decrease gradually from two end points to central point. Therefore, according to formulas (3.2-11), (3.2-16) and (3.2-17), we can conclude that the truncation error constants for higher order derivatives increase more quickly at points nearby boundary ends than at those around the vicinity of central point. Finally, it should be also pointed out that the present truncation error formulas and the respective conclusions are also applicable for the pseudo-spectral and collocation methods.

Table 3-1. The error constants in the present formulas for the truncation error under ten the equally spaced and the Chebyshev grid points

i	1	2	3	4	5	6	7	8	9	10
Chebyshev	5.83	0.17	0.013	0.015	0.02	0.028	0.04	0.08	0.03	6.14
Equally	4.89	0.32	0.049	0.011	0.001	0.004	0.02	0.06	0.36	5.29

Table 3-2. The maximum error constant (at end point N) in the present truncation error formulas

N	Equally spaced grids				Chebyshev grids			
	err(1)	err(2)	err(3)	err(4)	err(1)	err(2)	err(3)	err(4)
5	0.2	3.7	38.6	260.8	0.4	8.4	91.4	629.2
10	0.1	5.3	173	4148	0.075	6.1	264.8	7662
15	0.07	6	371	17109	0.013	2.4	229	15136
20	0.05	6.8	616	43704	0.002	0.7	119	14044

3.3. Simplified Formulas for Weighting Coefficients

In this section the formulas computing weighting coefficients for some often used grid spacings are discussed.

Civan and Sliepcevich (1984b) pointed out that the accuracies of the DQ solutions for some problems increased with the number of equally-spaced grid points up to 1/1. When the grid points were increased to 15 the accuracy dropped off. Civan and Sliepcevich (1984b) attributed the behaviors of the DQ method to inaccuracies of the DQ weighting coefficient caused by inversion computation of an ill-conditioning Vandermonde matrix (Hamming, 1973) during the evaluation of these coefficients. They suggested that an optimum number of grid points 11 should be used for equally spacings. The inaccuracies of the DQ weighting coefficient is one of tow factors which give rise to the unstable behaviors in the DQ method. Another is the grid spacing which will be discussed in the following section 3.4.

The effect of the ill-conditioning Vandermonde matrix was overcome due to recent work (Quan and Chang 1989a, Shu and Richards, 1992, Civan, 1989). Civan (1989) applied special algorithms in the solution of a Vandermonde system of equations (Bjorck and Pereyra, 1970) to obtain the exact DQ weighting coefficients. Quan and Chang (1989a) presented explicit formulas for the calculation of the DQ weighting coefficients. Shu and Richards (1992) also independently presented the same formulas and derived a useful recurrence formulas for weighting coefficients of higher order derivatives. It was proven that the BP algorithm generally yields the same accuracy of weighting coefficients as the explicit formulas (Malik and Civan, 1995, Bert and Malik, 1996). It is preferred to use the BP algorithms for computing weighting coefficients of the integral and composite operators (Bert and Malik, 1996), while the explicit formulas may be simple and convenient for computing the weighting coefficients of single differential operators.

Quan and Chang (1989a) also pointed out that the DQ method can use all zeros of the Chebyshev and Legendre polynomials as its grid points, and such sampling points can naturally encompass the boundary end points without additional forced points, while the orthogonal collocation method can not. Therefore, the DQ method using the grid spacing of such type has somewhat faster convergence rate than the orthogonal collocation method (1989b). Quan and Chang (1989a) also gave the direct computing formulas for the evaluation of the DQ weighting coefficients when using these grid spacings. However, these formulas are still somewhat complex and inconvenient for practical use. We herein give simpler and more explicit alternative computing formulas.

The roots of the N order Chebyshev polynomials in (-1, 1) domain are given by

$$r_i = \cos\frac{(2i-1)\pi}{2N}, \qquad i = 1,2,\ldots,N. \qquad (3.3\text{-}1)$$

The Chebyshev grid spacing for any finite range x∈[a, b] can be obtained by a simple algebraic transformation,

$$x_i = \frac{b-a}{r_N - r_1} r_i + \frac{ar_N - br_1}{r_N - r_1}, \quad \text{i=1, 2, ..., N.} \qquad (3.3\text{-}2)$$

Using the properties of the Chebyshev polynomials and after some manipulations, we can derive the direct computing formulas for the 1st order derivative under this grid spacing.

$$A_{ij} = \frac{r_N - r_1}{b-a} \frac{(-1)^{(i-j)}}{(r_i - r_j)} \sqrt{\frac{(1-r_j^2)}{(1-r_i^2)}}. \qquad i \neq j \qquad (3.3\text{-}3)$$

and

$$A_{ii} = \frac{1}{2}\frac{r_N - r_1}{b-a}\frac{r_i}{(1-r_i^2)}, \qquad i = 1,2,\ldots,N. \qquad (3.3\text{-}4)$$

The weighting coefficients for higher order derivatives can be easily computed by using recurrence formulas presented by Shu and Richards (1992), namely equation (1.2-4) in section 1.2. The formulas (3.3-3) and (3.3-4) are obviously simpler and easier to be used than those presented by Quan and Chang (1989a).

Similarly, the computing formulas for the weighting coefficients of the 1st order derivative using the roots of the shifted Legendre polynomials in domain [0, 1] (Eqs. (5) and (7) in Bellman, Kashef and Casti (1972)) can be modified to include the boundary point directly for any finite range [a, b] by multiplying a constant coefficient $\frac{\lambda_N - \lambda_1}{b - a}$, where λ_1 and λ_N are the first and last roots of the N order shifted Legendre polynomials. In addition, there exists a typographical error in Eq. (7) of that reference on page 43:

$$A_{ii} = \frac{1 - 2x_k}{2x_k\left(x_k - 1\right)} \tag{3.3-5}$$

The formula should be correctly stated as:

$$A_{ii} = \frac{1 - 2x_k}{2x_k\left(x_k^2 - 1\right)} \tag{3.3-6}$$

where x_k is the kth roots of the shift Legendre polynomials.

For the equally spaced grid points, Chen and Yu (1993a) also proposed the following simplified formulas for the 1st order derivative in the finite range [a, b]:

$$A_{ij} = \frac{\left(-1\right)^{j-i}}{b - a}\frac{N - 1}{i - j}\frac{C_{N-1}^{j-1}}{C_{N-1}^{i-1}}, \qquad i \neq j \tag{3.3-7}$$

$$A_{ii} = \frac{N - 1}{b - a}\sum_{j \neq i}^{N}\frac{1}{i - j} \tag{3.3-8}$$

where C denotes the combination computation.

In order to avoid the difficulty due to the fact that the zeros of the conventional shifted Legendre and shifted Chebyshev polynomials are not located at the boundary points, Bert et al. (1993) introduced extra grid points to satisfy the boundary conditions. The DQ method using the grid spacings of this type is equivalent to the orthogonal collocation method. For the analysis of plates (Chen, Yu and Wang, 1996b), the DQ method using the present Chebyshev grid spacing (e.g. equation 3.3.2) has faster rate of convergence than using both the equally spaced grid points and the grid points presented by Bert et al. (1993).

3.4. A Preliminary Study of Grid Spacings

Even if the accurate weighting coefficients are obtained, the application of various grid spacings in the DQ method will have a greatly different numerical stability behavior and convergence rate. It is shown that the DQ method using zeros of orthogonal polynomials are in general more efficient and reliable for some problems (Quan et al., 1989a, b; Bert et al., 1993). Quan and Chang (1989b) compared performances of a variety of existing grid spacings through the solution of some typical chemical engineering problems. Their conclusion is that the DQ method using the zeros of Chebyshev polynomials may be the most efficient. Similar to the polynomial interpolation approximation for a function, equally spaced grid points were in general convenient but obviously not reliable for some cases (Sherbourne et al., 1991; Wang et al., 1994a). Bert and Malik (1996) suggested that the so-called Chebyshev-Gauss-Lobatto points may be consistently better than the equally spaced, Legendre, and Chebyshev points in a variety of problems since such grid points encompass the boundary end points of the normalized domains. The task of this section aims to provide several general rules for determining the optimal grid spacings in the DQ method.

The following six types of sampling points are used in this study. All variable domains in the present study have been normalized as $x \in [0, 1]$.

Type I: Equally spaced sampling points

$$x_i = \frac{i - 1}{N - 1}, \quad i=1,2,....N. \tag{3.4-1}$$

Type II: Normalized Chebyshev sampling points

$$x_i = \frac{1}{2} - \frac{cos\,\frac{(2i-1)}{2N}\pi}{2\,cos\,\frac{\pi}{2N}} \quad \text{i=1, 2, ..., N.} \tag{3.4-2}$$

Type III: Equally spaced sampling points with δ points immediate adjacent two ends

$$x_1 = 0,\ x_2 = \delta,\ x_{N-1} = 1-\delta,\ x_N = 1$$

$$x_i = \frac{i-2}{N-3},\ i = 3, 4,\ \ldots,\ (N-2). \tag{3.4-3}$$

Type IV: Normalized Chebyshev sampling points with δ points immediate adjacent two ends

$$x_1 = 0,\ x_2 = \delta,\ x_{N-1} = 1-\delta,\ x_N = 1$$

$$x_i = \frac{1}{2} - \frac{cos\,\frac{(2N-5)\pi}{2(N-2)}}{cos\,\frac{\pi}{2(N-2)}},\ i = 3, 4,\ \ldots (N-2), \tag{3.4-4}$$

Type V: Equally spaced sampling points with only one δ point immediate adjacent one boundary end

$$x_1 = 0,\ x_2 = \delta,$$

$$x_i = \frac{i-2}{N-2},\ i = 3,\ \ldots N. \tag{3.4-5}$$

Type VI: Normalized Chebyshev sampling points with only one δ point immediate adjacent one boundary end

$$x_1 = 0,\ x_2 = \delta,$$

$$x_i = \frac{1}{2} - \frac{cos\,\frac{(2i-3)\pi}{2(N-1)}}{2\,cos\,\frac{\pi}{2(N-1)}},\ i = 3, 4,\ \ldots,\ N. \tag{3.4-6}$$

We choose N=7 and δ=10^{-5} in this study otherwise specified, the corresponding grid coordinates of the above six types in normalized domains are listed in table 3-3.

Table 3-3. Grid coordinates of the six types of sampling points (N=7 and δ=10^{-5})

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
Type I	0.0	0.167	0.333	0.5	0.667	0.833	1.0
Type II	0.0	0.099	0.277	0.5	0.723	0.901	1.0
Type III	0.0	0.00001	0.25	0.5	0.75	0.99999	1.0
Type IV	0.0	0.00001	0.19098	0.5	0.80902	0.99999	1.0
Type V	0.0	0.00001	0.2	0.4	0.6	0.8	1.0
Type VI	0.0	0.00001	0.13397	0.36602	0.63398	0.86603	1.0

The following four nonlinear differential equations serve as the numerical examples in a comparison study of the six types of sampling points. These examples are not very complex and not expected to be adequate in all circumstances. It is our opinion that the simple problems can provide some innovations into the more complex problems.

Example 1: $y' - e^{-y} = 0$; y(0)=0, 0≤x≤1 (3.4-7)

Example 2: $y'^2 + y^2 - 1 = 0$; y(0)=0, 0≤x≤1 (3.4-8)

Example 3: $y' - y^2 - 1 = 0$; y(0)=0, 0≤x≤1 (3.4-9)

Example 4: $y'' + \frac{1}{y} + \frac{y'^2}{y} = 0; \qquad y(0) = 1, y(1) = 2$ (3.4-10)

err$_i$ denotes the relative error at the ith grid point, e.g.,

$$err_i = \left| \frac{Analytic - Numerical}{Analytic} \right| \tag{3.4-11}$$

The average error is defined as

$$\bar{\varepsilon} = \frac{1}{N} \sum_{j=1}^{N} err_i \, . \tag{3.4-12}$$

The average relative errors for the above four examples are displayed in table 3-4. It is observed that the DQ method using different grid spacings yields approximate solutions of different accuracies. Based on comparison, the sampling points of type VI is the most accurate in the DQ solution of the first three examples, while type II produces the most accurate solutions for the fourth example. Obviously, types I, II, III and IV are symmetric spaced grid points, while types V and VI are not symmetric, e.g., where only one δ point is placed at immediate adjacent position of one boundary end. We can find that the asymmetric grid points seem to performs better for the problems with single boundary condition. In contrast, symmetric sampling points are more accurate for problems with double boundary conditions, one condition each with the two end points. This fact is also discovered again in applying the DQ method for analysis of deflection and vibration of beam and plates with two boundary conditions at each boundary. As shown in table 3-8, type IV gives the best solutions for these cases, while type VI yields the worst ones. In the case study for beam and plate, we exploit the technique proposed by Chen et al. (1993a), Du et al. (1994), Wang (1995), and Bert et al. (1996c) in applying multi-boundary conditions, namely, the so-called DQZ approach as defined in later section 4.4. The technique uses the boundary condition analog equation instead of the DQ analog of the governing equations at the boundary and its immediate adjacent points.

As were pointed out by Quan et al. (1989b) and Bert et al. (1994b), the Chebyshev-type points (e.g. type II, IV, and VI) in general produce more accurate solutions than the equally spaced points. It can be observed from table 3-3 that the Chebyshev grid spacing has a obvious tendency towards neighboring area of boundary ends than the equally spaced points. We define this phenomena as the boundary effect, namely, grid points should be placed to tend towards boundary ends where boundary conditions are impose. Although in this study we do not involve the Legender points and so-called Chebyshev-Gauss-Lobatto points, it is noted that the Chebyshev points in general have stronger tendency towards two ends and sparser density in central region than either the Legendre points or the Chebyshev-Gauss-Lobatto points. Quan and Chang (1989a, b) pointed out that the Chebyshev points may be most efficient sampling points based on some numerical experiments in chemical engineering area. The boundary effect provides an explanation for this conclusion.

Table 3-4 shows that the so-called δ spacings (types III, IV, V and VI) seem to be very effective for examples 1, 2 and 3 with single boundary condition. In these cases, the boundary effect manifests more obviously. Table 3-5 provides the relative errors at various grid points for example I in applying type II and IV of sampling points. The most inaccurate DQ solution is found at the second grid point for type II and at third grid point for type IV. This phenomena also occurs in using types I and III for example 2 and 3. The accuracies at δ points are obviously higher than those at other grid points. Table 3-9 compares the accuracies of type III with various δ value. δ=0.01 seems to be preferred. We find a most effective sampling points for the DQ analysis of static deflection of SS-SS-SS-SS square plate, which is shown in Table 3-10 and δ is arbitrary value within the scope of 0 to 0.181. The DQZ approach using such sampling points yields accurate solution 4.062×10^{-3} at center point, and 9.864 (relative error 0.06%, exact solution 9.869) for fundamental frequency of a SS-SS beam which is much more accurate than using the same number of any other existing grid spacing. A very interesting fact is also noted that the locations of δ points can not produce any affect on the DQ solutions of beams and plates when the DQZ technique is used to apply multi-boundary conditions in the DQ analysis of structural component analysis. The reason for this phenomena may be due to the fact that the DQZ approach applies the boundary conditions to replace the DQ analog equation of the governing equations at δ points immediate adjacent boundaries. In contrast, similar situations do not occur when the new approach in applying multiple boundary conditions proposed by Wang and Bert (1993a) is used. It is noted that the δ grid spacings are not applicable in vibrational analysis of beams and plates by means of Wang and Bert’s approach.

The absolute truncation error constants for the 1st and 2nd order derivatives in applying the present six types of sampling points are displayed in tables 3-6 and 3-7. It is noted that the truncation error constants are closely related to the grid spacings. In general, the error constants at points nearby boundaries are obviously larger than those at points nearby center region. The truncation error constants under the asymmetric grid spacings (types V and VI) is also asymmetric, while the symmetric sampling points yield basically symmetric distributions of error constants. The error constants for the 2nd derivative are also obviously larger than for the 1st order derivative. Also, it is a very interesting fact that smaller error constant at certain point can not ensure to obtain more accurate solution at the same point. The accuracies of the solution depend more on the total error distribution. However, it is worth stressing that the error constants at boundary points have not effect on the accuracies of the solutions if the DQ approximations are not required at the boundary points. The sample points, where error constants values have smaller relative difference, in general yield more accurate solutions, for example, the Chebyshev points. On the basis of the above case analysis, the following general rules for choosing sampling points in terms of efficiency and accuracy are concluded:

1) The Chebyshev points are usually the first choice for polynomial approximation of a function. However, such situations do not occur in numerical approximation of partial derivative of a function. No grid spacing is in general consistently more efficient than any other possible one for differential equations of different types.
2) The choice of sampling points is deeply related to the boundary conditions for problems of interest. For double boundary conditions, one each at two ends, the symmetric spaced grid points are preferred. In contrast, the asymmetric sampling points may be more effective in situations of problems with the single boundary condition.
3) In order to avoid the so-called boundary effect, sampling points should be more inclination to boundary ends where boundary conditions are imposed.
4) The accuracies of the DQ approximation using certain type of sampling points are determined to great extent by the relative difference among truncation error constants at various inner points including ends where no boundary conditions are imposed. In general, the sampling points, whose truncation error constants varied in a smoother form, yield more accurate approximation. Moreover, only the truncation errors of the highest order derivative in differential governing systems need be considered.

5) The δ value in the so-called δ spacings has not effect on the accuracies of the DQZ approach for analysis of beams and plates. Therefore, the choice of grid spacings for the DQZ need not consider the location of δ points, namely, if N points are used, we only consider the spacing problem of N-2 sampling points. However, it is noted that δ points must locate between the boundary points and adjacent inner points.
6) The δ spacings is in general more effective for the DQ solution of differential equations.
7) Rule 2 is more important than rule 4 in choosing sampling points.

Since the DQ method and the collocation method are equivalent except for the different choices of grid points (Quan and Chang, 1989a), the above rules are also applicable for the collocation method. It is observed that the high efficiency using a kind of grid spacing for some problems usually means better stability and reliability. The present study only consider the DQ method based on the usual polynomial basis functions. It should be pointed out that the DQ method using other types of basis functions, which may be more proper for certain problems, have very fast speed of convergence and strong stable, namely, no limitation on the number of grid points.

Table 3-4. A comparison of average errors of the DQ solutions using different sampling points (δ=0.00001).

	Type I	Type II	Type III	Type IV	Type V	Type VI
Example I	1.8E-4	5.7E-5	4.0E-5	2.1E-5	1.96E-5	9.98E-6
Example II	2.1E-6	1.1E-6	5.8E-7	1.7E-7	2.6E-7	1.3E-7
Example III	1.6E-2	8.4E-3	9.0E-3	3.7E-3	1.4E-3	6.4E-4
Example IV	2.1E-4	3.4E-5	2.2E-4	1.6E-4	1.2E-3	1.5E-4

Table 3-5. Accuracy comparison for example 1 at various grid points

	2	3	4	5	6	7
Type 2	2.2E-4	5.1E-5	3.2E-5	1.9E-5	1.6E-5	1.3E-5
Type 4	2.9E-8	1.3E-4	4.2E-5	3.3E-5	1.9E-5	1.9E-5

Table 3-6. Absolute value of truncation error constants for the first order derivative

Grids	1	2	3	4	5	6	7
Type I	3.6E-1	6.0E-2	2.4E--2	1.8E-2	2.4E-2	6.0E-2	3.6E-1
Type II	2.1E-1	7.4E-2	5.2E-2	4.7E-2	5.2E-2	7.4E-2	2.1E-1
Type III	2.2E-5	2.2 E-5	1.0 E-1	9.1E-2	1.0E-1	2.2E-5	2.2E-5
Type IV	1.8E-5	1.8E-5	1,1 E-1	1.4E-1	1.1E-1	1.8E-5	1.8E-5
Type V	8.9 E-8	8.9E-6	3.5E-2	3.5E-2	5.4 E-2	1.9E-1	8.9E-1
Type VI	6.3 E-6	6.3E-6	3.1E-2	6.2E-2	1.1E-1	2.0E-1	6.3 E-1

Table 3-7. Absolute values of truncation error constants for the second order derivative

Grids	1	2	3	4	5	6	7
Type I	9.9	8.1E-1	1.2E-1	3.6E-2	2.2E-1	1.0	11.3
Type II	7.6	1.4E-1	4.9E-2	9.3E-2	1.6E-1	4.4E-1	8.4
Type III	4.4	4.4	6.6E-2	1.8E-1	3.4E-1	4.4	4.4

Type IV	3.6	3.6	8.8E-1	2.8E-1	4.6E-1	3.6	3.6
Type V	1.8	1.8	4.2E-2	1.3E-1	4.7E-1	2.2	24.1
Type VI	1.3	1.3	4.4E-1	4.3E-1	6.1E-1	1.4	20.0

Table 3-8. Relative errors of DQ solutions of center deflections of square plates

	Type I	Type II	Type III	Type IV	Type V	Type VI
SS-SS-SS-SS	2.6E-2	1.9E-2	1.4E-2	2.5E-3	9.6E-2	1.6E-1
C-C-C-C	2.4E-2	1.2E-2	5.5E-3	7.9E-3	1.3E-1	2.3E-1

Table 3.-9. The effect of δ values on DQ average errors using type III grid spacing

δ	Example 1	Example 2	Example 3	Example 4
0.1	5.7E-5	1.0E-6	8.8E-3	1.1E-4
0.075	4.5E-5	8.1E-7	6.2E-3	1.4E-5
0.01	3.9E-5	6.4E-7	7.8E-3	2.3E-4
0.001	3.9E-5	6.6E-7	8.7E-3	2.3E-4
0.0001	4.0E-5	6.9E-7	8.9E-3	2.2E-4
0.00001	4.9E-5	5.8E-7	9.0E-3	2.2E-4

Table 3-10.

0.0	δ	0.181	0.5	0.819	1-δ	1.0

CHAPTER 4

BOUNDARY VALUE PROBLEMS

3.1. Introduction

Chen, Wang and Zhong (1996b) proposed the new DQ approximate formulas in matrix form for partial derivatives in tow-dimensional domain, which were different from the conventional ones presented by Civan and Sliepcevich (1984b). By using these new approximate formulas, the DQ formulations for the Poisson and convection-diffusion equations can be expressed as the Lyapunov algebraic matrix equation. The formulation effort is greatly simplified, and a simple and explicit matrix formulation is obtained. A variety of fast algorithms in the solution of the Lyapunov equation (Bartels and Stewart, 1972; Golub et al., 1979; Gui, 1992) can be successfully applied in the DQ analysis of these two-dimensional problems, and, thus, the computing effort and storage requirements can be greatly reduced. Finally, we also point out that the present reduction technique can be easily extended to the three-dimensional cases.

Many applications has shown that the DQ method is a very efficient technique for structural analysis. Bert and Malik (1996b) proposed a semi-analytical approach for further improving the efficiency of the DQ technique for some structural problems. However, the conventional applications of the DQ method does not work well. Wang and Bert (1993a) presented a new approach for application of multiple boundary conditions and proved that it was more efficient than the conventional approach for the analysis of some structural component. However, this new approach is not equally successful for beams and plates with clamped-clamped (C-C) boundary condition and other complex conditions. Chen et al. (1993, 1994) and Du et al. (1994) presented a different approach for the same task. The efficiency and simplicity of the approach are demonstrated via numerical examples. We also gave a theoretical explanation for this method. We compare these three approaches through numerical experiments.

The geometrically nonlinear bending of orthotropic plate were recalculated by the DQ method (Chen, Zhong and He, 1996h). Due to the application of the Hadamard product and SJT product techniques and new matrix approximation formulas, the formulation and programming effort is greatly simplified, the computing effort and storage requirements are reduced to about one-twenty seventh and one-ninth as much as those in Bert et al. (1989). New approaches for applying boundary conditions also improve the accuracies of the DQ solutions for this case.

4.2. New Approximate Formulas and Lyapunov Matrix Equation

Chen et al. (1996b) gave the following DQ formulation in matrix form for the partial derivative of the function ψ(x,y) in two-dimensional variable domain, e.g.

$$\frac{\partial^2 \psi}{\partial x^2} = B_x \psi, \qquad \frac{\partial^2 \psi}{\partial x \partial y} = A_x \psi A_y^T, \qquad \frac{\partial^2 \psi}{\partial y^2} = \psi B_y^T,$$
$$\frac{\partial \psi}{\partial x} = A_x \psi, \qquad \frac{\partial \psi}{\partial y} = \psi A_y^T \tag{4.2-1}$$

where the unknown ψ is a rectangular matrix rather than a vector as in other references. The matrices A and B with subscripts stand for the DQ weighting coefficient matrices for the 1st and 2nd order partial derivatives with respect to the corresponding independent variables, respectively. The superscript T on the A and B means the transpose of the matrices. There are similar approximate formulas for higher order derivatives. Formula (4.2-1) is very useful to reduce the formulation and computational effort in practice

(Chen, Zhong and He, 1996h; Chen and Zhong, 1996i). It is noted that the present DQ approximate formulas in matrix form can be easily extended to three-dimensional problems.

In terms of the matrix approximate formulas (4.2-1), the governing equations for some boundary value problems can be easily converted into a Lyapunov algebraic matrix equation. The detailed examples will be provided in the later section 4.3. In what follows we discuss the several existing fast algorithms for the solution of the Lyapunov equation.

The Lyapunov matrix equation are often encountered in the optimal control, and several efficient methods for solving equations of such type have been proposed. To simplify the presentation, BS, HS and R-THR represents the methods presented, respectively, by Bartels and Stewart (1972), Golur , Nash and Loan (1979), and Gui (1992). In the following we only discuss the computational efficiency of these methods. Detailed information on their use see the corresponding references. All these methods are also stable and in general include the following four steps (Nash and Loan, 1979, and Gui, 1992). For example, considering the Lyapunov equation

$$\psi G + R\psi = Q \ , \qquad (4.2\text{-}2)$$

where G, R and Q are constant rectangular matrix, ψ is the rectangular matrix composed of the desired values. We have

Step 1: Reduce G and R into certain simple form via the similarity transformations $G^*=P^{-1}GP$ and $R^*=S^{-1}RS$.

Step 2: $F= S\,Q\,P^{-1}$ for the solution of F.

Step 3: Solve the transformed equation $V\,G^*+ R^*V=F$ for V.

Step 4: $\psi=SVP^{-1}$.

The respective computational effort is listed in table I (Gui, 1992).

Table 4-1. Comparison of computational effort in the BS, HS and R-THR methods

Steps Methods	$G^*=P^{-1}GP$ $R^*=S^{-1}RS$	$F= S^{-1}QP$ for F	$VG^*+R^*V=F$ for V	$\psi=SVP^{-1}$ for ψ
BS	$10(m^3+n^3)$	m^2n+mn^2	$(m^2n+mn^2)/2$	m^2n+mn^2
HS	$5m^3/3+10n^3$	m^2n+mn^2	$3m^2n+mn^2/2$	m^2n+mn^2
R-THR	m^3+n^3	m^2n+mn^2	$n^3/3+5m^2n+ 3mn^2$	m^2n+mn^2

Based on consideration of the computational effort, the R-THR method is the most efficient for boundary value problems. The detailed analysis on the computational efficiency can be found in the latter section 4.3.

On the other hand, the DQ formulations for some problems are not Lyapunov matrix equation, even if the present matrix approximate formulas are applied, for example, problems in field of structural mechanics. However, the use of the present approximate formulas can greatly reduce the formulation effort and yield a explicit matrix formulation. Therefore, the programming is simplified significantly. The desired ψ in rectangular matrix form can be converted into the conventional vector form by the following Lemma 4.2.1.

Lemma 4.2.1. If $A\in C^{p\times m}$, $B\in C^{n\times q}$ and the unknown $X\in C^{m\times n}$, then

$$vec(AXB)=(A\otimes B^T)vec(X) \tag{4.2-3}$$

where vec( ) is the vector-function of a rectangular matrix formed by stacking the rows of matrix into one long vector, ⊗ denotes the Kronecker product of matrices. To simplify the presentation, we define $vec\left(AXB\right)=AX\vec{B}$ and $vec\left(X\right)=\vec{X}$.

Corollary:

$$\begin{aligned}&1\ \ A\vec{X} = \left(A\otimes I_n\right)\vec{X}\\ &2\ \ X\vec{B} = \left(I_m\otimes B^T\right)\vec{X}\\ &3\ \ A\vec{X}+X\vec{B} = \left(A\otimes I_n+I_m\otimes B^T\right)\vec{X}\end{aligned} \tag{4.2-4}$$

where I_n and I_m are the unit matrix.

The examples on anisotropic plate in section 6.3 and geometrically nonlinear plates in section 4.5 show obvious advantages of the present matrix approximate formulas.

4.3. Convection-Diffusion Equations and Poisson Equation

Compared with the Galerkin, Control-volume and finite difference methods, the differential quadrature (DQ) method has proved to be a most efficient numerical technique in the calculation of the Poisson and convection-diffusion equations (Civan and Sliepcevich, 1983a, b; 1984b). The present study deals with further improvement of efficiency of the DQ method for these cases. By using the new approximate formulas presented in section 4.2, the DQ formulations for the Poisson and convection-diffusion equations can be expressed as the Lyapunov algebraic matrix equation. The formulation effort is simplified, and a simple and explicit matrix formulation is obtained. A variety of fast algorithms in the solution of the Lyapunov equation can be successfully applied in the DQ analysis of these two-dimensional problems, and, thus, the computing effort and storage requirements can be greatly reduced. Finally, we also point out that the present reduction technique can be easily extended to the three-dimensional cases (Chen et al., 1996i).

4.3.1. Formulations in the Lyapunov Matrix Equation Form

Unlike the conventional DQ method (Civan and Sliepcevich, 1983a, b; 1984b), the DQ weighting coefficients here are modified in advance by using the boundary conditions. For example, considering the Dirichlet and Neumann boundary conditions in the x-direction (x∈[0,1])

$$\phi\left(0,\ y\right) = h \tag{4.3-1}$$

$$\frac{\partial\phi\left(1,\ y\right)}{\partial x} = q\ . \tag{4.3-2}$$

Eq. (4.3-2) can be approximated by

$$\sum_{j=1}^{N}A_{Nj}\,\phi_j = q\ . \tag{4.3-3}$$

The function values at boundary points can be expressed by the unknown interior point function values, namely,

$$\phi_N = \frac{1}{A_{NN}}\left(q - A_{N1}h - \sum_{j=2}^{N-1} A_{Nj}\,\phi_j\right). \tag{4.3-4}$$

Substituting equations (4.3-4) and (4.3-1) into the DQ formulations for the first and second derivatives, respectively, we have

$$\frac{\partial\vec{\phi}}{\partial x} = \bar{A}_x\vec{\phi} + \vec{a}_x \tag{4.3-5}$$

$$\frac{\partial^2\vec{\phi}}{\partial x^2} = \bar{B}_x\vec{\phi} + \vec{b}_x, \tag{4.3-6}$$

where $\vec{\phi} = \{\phi_2 \quad \phi_3, \quad \ldots, \quad \phi_{N-1}\}$, $\vec{a}_x$ and $\vec{b}_x$ are the constant vectors, $\bar{A}$ x and $\bar{B}$ x are the modified (N-2)×(N-2) weighting coefficient matrices for the 1st and 2nd order derivatives, respectively. Also, it is noted that the similar procedures can be used to incorporate any complex linear boundary conditions into the DQ weighting coefficient matrices. Substituting Eqs. (4.3-5) and (4.3-6) into matrix approximate formulas (4.3-1), we have

$$\frac{\partial^2\tilde{\phi}}{\partial x^2} = \bar{B}_x\tilde{\phi} + B_{ox}, \qquad \frac{\partial^2\tilde{\phi}}{\partial y^2} = \tilde{\phi}\bar{B}_y^T + B_{0y}^T, \qquad , \tag{4.3-7}$$
$$\frac{\partial\tilde{\phi}}{\partial x} = \bar{A}_x\tilde{\phi} + A_{ox}, \qquad \frac{\partial\tilde{\phi}}{\partial y} = \tilde{\phi}\bar{A}_y^T + A_{oy}^T$$

where the unknown $\tilde{\phi}$ is a n×m rectangular matrix rather than a vector as in Civan and Sliepcevich (1983a, b, 1984b), n and m is the number of inner grid points along x- and y- directions, respectively. The superscript T means the transpose of the matrices. A_{0x} and B_{0x} are generated by stacking the corresponding constant vectors $\vec{a}_x$ and $\vec{b}_x$ in Eqs. (4.3-5) and (4.3-6). For example,

$$A_{0x} = \begin{bmatrix} a_{1x} & a_{1x} & \cdots & a_{1x} \\ a_{2x} & a_{2x} & \cdots & a_{2x} \\ \vdots & \vdots & \ddots & \vdots \\ a_{nx} & a_{nx} & \cdots & a_{nx} \end{bmatrix}_{n\times m}. \tag{4.3-8}$$

A_{0y} and B_{0y} can be obtained in a similar way. For higher order partial derivatives, there exist similar matrix approximate formulas.

The Poisson equations can be normalized as:

$$\frac{\partial^2\phi}{\partial x^2} + \beta^2\frac{\partial^2\phi}{\partial y^2} + S = 0, \tag{4.3-9}$$

where x and y are the dimensionless Cartesian coordinates, namely x, y∈[0, 1], β denotes the aspect ratio, S is a given strength, ϕ is the desired variable. More details see Civan and Sliepcevich (1983b).

Applying the matrix approximate formulas (4.3-7), the DQ formulation for equation (4.3-9) is given by

$$\bar{B}_x\tilde{\phi} + \beta^2\tilde{\phi}\bar{B}_y^T + H = 0, \tag{4.3-10}$$

where $\tilde{\phi}$, $\bar{B}_x$ and $\bar{B}_y$ are (n-2)×(n-2) rectangular matrix, $H = S + B_{0x} + B_{oy}^T$. Since the boundary conditions have been taken into account in the formulation of weighting coefficient matrices $\bar{B}_x$ and $\bar{B}_y$, no additional equations are more required.

The equation governing steady-state convection-diffusion (e.g., equation (24) in Civan and Sliepcevich (1983b) neglecting time derivative term) can be simplified as

$$\alpha\frac{\partial\varphi}{\partial x}+\beta\frac{\partial\varphi}{\partial y}=\frac{\varphi}{4\alpha},\tag{4.3-11}$$

where φ is the desired values as defined in equation (23) in Civan and Sliepcevich (1983b), α and β are constants. In terms of the new DQ matrix approximate formulas (4.3-7), we have

$$\alpha\overline{B}_x\varphi+\beta\varphi\overline{B}_y^T-\frac{1}{4a}\varphi=Q\ ,\tag{4.3-12}$$

where Q is constant matrix generated from the modified DQ weighting coefficient matrices similar to H matrix in Eq. (4.3-10). Furthermore, the above equation can be restated as

$$\left(\alpha\overline{B}_x-\frac{1}{4a}\right)\varphi+\beta\varphi\overline{B}_y^T=Q\tag{4.3-13}$$

The above DQ formulations (4.3-11) and (4.3-13) are the Lyapunov algebraic matrix equation. Obviously, they are more explicit and simpler than the conventional polynomial formulations given in Civan and Sliepcevich (1983a, b; 1984b).

4.3.2. Results

The respective computational effort for the algorithms of solution of Lyapunov equation is listed in table 4-I in section 4.2. The total computing effort in these methods is O(n^3+m^3) scalar multiplications. The R-THR method requires $n^3+\frac{4}{3}m^3+7n^2m+5mn^2+n^2$ (or $14\frac{1}{3}n^3+n^2$ when n=m) scalar multiplications, and may be the most efficient in the solution of the Lyapunov matrix equations for the present purpose. By using the R-THR method, the same examples given in Civan and Sliepcevich (1983b) are recalculated by the DQ method, and the accuracies of results are coincident with those given by Civan and Sliepcevich (1983b). However, the conventional approach required solving a linear simultaneous equations of (N_x-2)(N_y-2) order by using the Gaussian elimination method, where N_x and N_y are the number of gird points along x- and y- directions, respectively. If N=N_x=N_y, about $\frac{1}{3}(N-2)^6$ multiplications were performed. In contrast, the present reduction approach requires about $14\frac{1}{3}(N-2)^3$ multiplications. Thus, the computational effort is only about 34% in using 7×7 grid points and 6% in using 11×11 grid points as much as that in Civan and Sliepcevich (1983b). The steady-state convection-diffusion (example 1 in Civan and Sliepcevich (1984b)) is also computed by using the present technique, and the same computing reduction is achieved. Gui (1992) also pointed out that the parallel computation was very efficient in the solution of the Lyapunov equations. It is well known that more grid points, more accurate DQ results, while the computational effort in the present DQ method for these cases is reduced in proportional to $(N-2)^3$.

4.3.3. On the three-dimensional problems

For three-dimensional cases, we first convert into it into a set of ordinary differential equations by using the new DQ matrix approximate formulas (4.3-7) and the Kronecker product. It is straightforward that the DQ matrix approximate formula for a set of ordinary differential equations is similar to formulas (4.3-7). Thus, the ordinary differential equations can be formulated into a Lyapunov matrix equation. The following examples can illustrate our idea more clearly. Considering the three-dimensional steady-state convection-diffusion equation (equation (52) in Civan and Sliepcevich (1984b) neglecting time derivative term)

$$\frac{\partial c}{\partial x} = \beta \frac{\partial^2 c}{\partial y^2} + \gamma \frac{\partial^2 c}{\partial z^2}. \qquad (4.3\text{-}14)$$

First, in terms of the DQ matrix approximate formulas (4.3-14), the above equation can be approximated as the following ordinary equations,

$$\bar{A}_x \tilde{C} - \beta \tilde{C} \bar{B}_y^T + Q = \gamma \frac{d^2 \tilde{C}}{dz^2}, \qquad (4.3\text{-}15)$$

where $\tilde{C}$ is a (N$_x$-2)×(N$_y$-2) rectangular matrix, Q=$A_{0x} - \beta B_{0y}^T$. By using the Kronecker product of matrices (Lancaster and Timenetsky), we have

$$\gamma \frac{d^2 \vec{C}}{dz^2} = \left[\bar{A}_x \otimes I_y - \beta I_x \otimes \bar{B}_y\right]\vec{C} + Q, \qquad (4.3\text{-}16)$$

where $\vec{C}$ is a ((N$_x$-2)(N$_y$-2)) ×1 vector stacked from matrix C. The DQ matrix approximate formula for the above ordinary differential equations can be written as

$$\left[\bar{A}_x \otimes I_y - \beta I_x \otimes \bar{B}_y\right]\hat{C} - \gamma \hat{C} \bar{B}_z^T = R \qquad (4.3\text{-}17)$$

where $\hat{C}$ is a ((N$_x$-2)N$_y$-2))×(N$_z$-2) rectangular matrix, R=γB_{oz}^T –Q. The above equation is also a Lyapunov matrix equation. Thus, the reduction technique for the Lyapunov equation can be used to achieve a considerable savings in computational effort.

4.3.4. Remarks

Civan and Sliepcevich (1983b; 1984b) suggested that special matrix solver should be developed to reduce the computing effort in applying the DQ method to the Poisson and convection-diffusion equations. The present work realizes this goal to minimizes the computational effort. It is shown that the principal advantages of the new matrix approximate formulas are to offer a more compact and convenient procedure for obtaining an explicit matrix formulation and make the DQ method computationally more efficient for multi-dimensional problems by means of the existing techniques in the solution of the Lyapunov equations. It is concluded that the presented DQ approximate formulas in matrix form are competitive alternatives to the conventional ones presented by Civan and Sliepcevich (1984b). The extension of the present reduction DQ method to the transient convection-diffusion equations are a current subject of further study.

4.4.1. Traditional DQN Approach

To apply the DQ method to the analysis of structural components, the traditional DQN approach required a special nonuniform grid spacing, in which two points, separated by a very small distance δ, are placed at each boundary point. One boundary condition each is imposed at the so-called δ point immediate adjacent boundary, and thus the approach can not satisfy exactly all boundary conditions and the accuracies of the solution are affected. The approach has been high efficiency for the C-C boundary condition but not equally successful for other boundary conditions (Wang and Bert, 1993a; Wang, Bert and Striz, 1993b). The accuracy of the solutions depends on the proper choice of δ. However, too small δ will result oscillation and deteriorate the computations. δ is usually determined by trial and error for different cases. This is a rather tedious work in practice. In addition, the number of grid points in the DQN approach can not be large due to the instability caused by the δ effect. The approach has been utilized to analyze a variety of structural components (Jang, 1989; Bert et al., 1988a, 1989; Farsa et al., 1991; Feng, 1992; Kukretic et al., 1992; Laurra, 1993, 1994a,b, 1996; Sherbourne et al. 1991; Striz et al., 1988), and extended to handle truss and frame structures (Striz et al., 1994a)

4.4.2. Wang and Bert's New Approach

To overcome the drawbacks in the DQN approach, Wang et al. (1993a, b, c, 1994a, b, 1995) and Bert et al. (1993) developed a new DQU approach. The essence of the technique is that the boundary conditions are applied during formulating the weighting coefficient matrices for inner grid points. The accuracy and efficiency using the DQU approach is much higher than those using the conventional DQN approach. However, it is regret that the DQU approach is not applicable for the C-C boundary condition. Another shortcoming is that the DQU approach seems not to be applicable for the problems with complex boundary conditions or extra constrains. Thus, a combination of the DQU and DQN approaches was used to handle the problems with both the C-C boundary condition and other boundary conditions in Wang et al. (1993b).

If Wang and Bert's new approach is employed to modify the weighting coefficient matrices in advance, the matrix approximate formulas (4.2-1) can be expressed as

$$\frac{\partial^4 \hat{\psi}}{\partial x^4} = \overline{D}_x \hat{\psi}, \qquad \frac{\partial^4 \hat{\psi}}{\partial x^3 \partial y} = \overline{C}_x \hat{\psi} \overline{A}_y^T, \qquad \frac{\partial^4 \hat{\psi}}{\partial x^2 \partial y^2} = \overline{B}_x \hat{\psi} \overline{B}_y^T,$$

$$\frac{\partial^4 \hat{\psi}}{\partial x \partial y^3} = \overline{A}_x \hat{\psi} \overline{C}_y^T, \qquad \frac{\partial^4 \hat{\psi}}{\partial y^4} = \hat{\psi} \overline{D}_y^T, \qquad \frac{\partial^2 \hat{\psi}}{\partial x^2} = \overline{B}_x \hat{\psi}, \qquad \frac{\partial^2 \hat{\psi}}{\partial y^2} = \hat{\psi} \overline{B}_y^T \tag{4.4-1}$$

where the unknown $\hat{\psi}$ is a rectangular matrix composed of desired function values at inner grid points, $\overline{A}$, $\overline{B}$, $\overline{C}$ and $\overline{D}$ with subscripts x and y here stand for the DQ weighting coefficient matrices, modified by the respective boundary conditions using Wang and Bert's DQU approach, for the 1st, 2nd, 3rd and 4th order partial derivatives, respectively.

4.4.3. Another New Approach

Chen (1994) noticed the fact that the rank of the weighting coefficient matrix for the ith order derivative is m-i, where m is the number of grid points. Moreover, the coefficient matrix is in fact a nilpotent matrix, namely the weighting coefficient matrix is a zero matrix when m=i. Therefore, the coefficient matrices in the DQ method must be modified into full rank matrices before practical computation. Chen and Yu (1993)[1] proposed a different DQZ approach to eliminate the δ effect in the conventional DQN approach. The rank of the DQ coefficient matrix for the 4th order derivative is N-4. Therefore, the DQ analog equations of the governing equations in the points immediate adjacent boundary ends are replaced by boundary equations, and all boundary conditions are imposed at boundary points exactly. Bert and Malik (1996d) emphasized that in the DQZ method the replacement of the quadrature analog equations at inner points can not be arbitrary for vibrational analysis of beams and plates. They also proposed to employ the δ grid spacings in the present DQZ approach to avoid that the inner points of invoking the boundary conditions are of the ones of zero displacements, i.e., the nodal points of the vibrating beams and plates. However, according to our work in section 3.4, the δ value of the δ grid spacings has no effect on the DQZ solutions of both static and vibration problems. The improvement on accuracies for example 3 using the δ grid spacings in Bert and Malik (1996d) is due to the application of the different grid spacings at inner sampling points[2] rather than the very samll δ value. Therefore, it seem not to be necessary to use the very small δ value in the DQZ approach.

[1] Du et al. (1994) also presented independently this approach in analyzing the structural components.

[2] Here inner sampling points exclude end points and their immediate adjacent points.

The DQZ approach is conceptually simple and effective for the C-C boundary conditions as well as any other boundary condition. The principal advantages of the approach are better numerical stability and somewhat higher accuracy in comparison to the conventional DQN approach as well as the DQU approach for many cases. The requirements for very small δ is not necessary as in the DQN approach. Therefore, the larger number of grid points can be used in the DQZ approach. Moreover, there is more room for selecting types of sampling points in the DQZ method. In this paper, an improvement is also made for the DQZ approach, namely, we eliminate the first two and last two rows of the original weighting coefficient matrix and apply the four boundary condition equations to modify it into (N-4)×(N-4) matrix of full rank before the formulation of two-dimensional partial differential systems. Bert and Malik (1996d) pointed out that two different boundary conditions at the same corner point from the respective two edges may cause one problem in the implementation of multiple boundary conditions. The so-called corner difficulty is avoided in the present DQZ formulation, since the boundary condition equations are incorporated into the DQ weighting coefficient matrices in advance. The resulting size of the formulation equations is also reduced. For example, the DQ approximate equations for the C-C boundary condition are given by

$$w_1 = 0, \qquad \sum_{j=1}^{N} A_{1j} w_j = 0 \tag{4.4-2}$$

and

$$w_N = 0, \qquad \sum_{j=1}^{N} A_{Nj} w_j = 0, \tag{4.4-3}$$

where w_j's are the corresponding displacement at the jth grid point. It is noted here that all boundary conditions are exactly satisfied at boundary points. The desired displacement at the 2nd and (N-1)th grid points are expressed in terms of the unknown function values at interior point, namely,

$$w_2 = -\frac{1}{A_{12}} \sum_{j=3}^{N-1} A_{1j} w_j \tag{4.4-4}$$

$$w_{N-1} = -\frac{1}{A_{N,N-1}} \sum_{j=2}^{N-2} A_{Nj} w_j \tag{4.4-5}$$

Substituting equations (4.4-4) and (4.4-5) into the DQ analog equations for the 1st, 2nd, 3rd and 4th derivatives at the 3rd, 4th until (N-2)th grid points, respectively, we have

$$\frac{d\vec{w}}{dx} = \bar{A}\vec{w}, \qquad \frac{d^2\vec{w}}{dx^2} = \bar{B}\vec{w}, \qquad \frac{d^3\vec{w}}{dx^3} = \bar{C}\vec{w}, \qquad \frac{d^4\vec{w}}{dx^4} = \bar{D}\vec{w} \tag{4.4-6}$$

where $\vec{w} = \{w_3, w_4, \dots, w_{N-2}\}$. $\bar{A}$, $\bar{B}$, $\vec{C}$ and $\bar{D}$ are (N-4)×(N-4) modified coefficient matrices different from those of (N-2)×(N-2) dimension in the DQU approach. For other boundary conditions, the modified coefficient matrices can be obtained in the similar way. The extension of this idea to the DQN approach is also beneficial to simplify its use. Matrix approximate formulas in the DQZ approach is similar to formulas (4.4-1) for the DQU approach.

4.4.4. Some Examples and Discussions

In the following, DQZ is first applied to solve the deflection of a cantilever beam under a centralized load at free end. In terms of the DQ method, the governing equation for this case can be expressed as

$$\sum_{j=1}^{N} D_{Nj} w_j = pL^3 / EI \tag{4.4-7}$$

and

$$\sum_{j=1}^{N} D_{ij} w_j = 0, \qquad i = 3,4,\ldots,N-3. \tag{4.4-8}$$

Applying the DQ method to the Boundary conditions in the DQZ way, we obtain

$$\begin{aligned} &w_1 = 0, && \sum_{j=1}^{N} A_{ij} w_j = 0 \\ &\sum_{j=1}^{N} B_{Nj} w_j = 0, && \sum_{j=1}^{N} D_{Nj} w_j = 0 \end{aligned} \tag{4.4-9}$$

It is obvious that the DQZ approach satisfies all boundary conditions at boundary ends. The analytic solution for this case is a third order polynomial, while the DQ method is a polynomial fitting method. The DQZ method gives analytic solutions for this case by using only five equally spaced grid points. Bert et al. (1994b) computed this case by the DQU method and obtained analytic results at all grid points except for solution at the loading end. We list all these results in table 4-2. Although the larger number of grid points (N=7, 9) is used, the DQU method can still not obtain analytic solution at loading end. Bert et al. (1994b) also used non-uniform grid spacing that a grid point is placed close to the loading end. The closer this grid point is to the loading end, the better the DQU approach yields solution at the loading end. But it should be pointed out that the analytical solution at the loading end can never be obtained by DQU. The DQU method has an inherent discontinuity for structures under concentrated load at boundary point. Applying the special grid spacing can decrease the effect of this discontinuity but not eliminate it. Similarly, consider the deflection of the circular plate under a concentrated load at center point, DQZ can obtain the exact solutions at all grid points including loading point, while DQU cannot. Therefore, the DQU approach appear not to be very suitable for the structural components subjected to the centralized load which are often encountered in practice.

Table 4-2. Nondimensionalized deflection w of a cantilever beam subjected to a concentrated end load P ($\overline{w} = w\,EI/PL^3$).

x/L	0	1/4	1/2	3/4	1.0
DQU (N=5)	0	0.028646	0.10417	0.21094	0.32884 (-1.3%)
(N=9)	0	0.028646	0.10417	0.21094	0.33300 (-0.1%)
DQZ (N=5)	0	0.028646	0.10417	0.21094	0.33333
EXACT	0	0.028646	0.10417	0.21094	0.33333

The DQU method is also not amenable to a C-C beam or plates with a combination of C-C and other boundary conditions. Wang (1993b) still compute this sort of problems by means of a combined use of DQN and DQU. We also apply DQZ (five grid points used) to the deflection of C-C beam under the uniformly distributed load, and analytic solutions are obtained at all grid points. In contrast, the deflection at center point by DQN (Wang et al., 1993b) was 0.0026031(δ=0.0005), while analytic solution is 0.0026042. Although the DQN error was very small, it was not exact solution. In fact, DQN cannot obtain analytical solution at any grid point for this case and the above cantilever beam because it can not exactly satisfy all boundary conditions at the boundary points.

The cases that the DQ method can give their analytic solutions are few. Consider the fundamental frequency of free vibration of a cantilever beam, the analytical solution is 3.516, the results by DQN is 3.524 (N=9, error 0.24%), DQU 3.514 (N=7, error -0.05%) in Wang et al. (1993b), and DQZ 3.517 (N=9, error 0.03%) in Bert and Malik (1996). Here N is the number of grid points, and equally spaced grid points with the δ points are used for DQZ and DQN. DQU uses the equally spaced points without the δ points. It is noted that the computing effort in the DQZ and DQN methods using N sampling points is basically equal to that in the DQU method using N-2 sampling points. Obviously, DQZ is the best approach for this case, while the accuracy of DQN is the worst. The above-mentioned examples are provided to demonstrate that DQZ is more effective than DQU and DQN in many cases. So far we have not found cases for which DQN is more effective than DQZ. DQN has been shown to be very efficient for analysis of clamped beam and plate (Wang et al, 1994; Bert and Malik, 1996). Even if considering these cases, the DQZ approach also has the same computing accuracies and efficiency as the DQN approach if the DQZ approach applies the similar grid spacings. Therefore, it is concluded that the DQZ approach is very competitive alternative to the DQN approach. The conclusion is also supported by the following case analysis. However, it should be pointed out that DQU can yield better results for some problems than DQZ in a comparable computational effort. For instance, considering the simply supported beam, the relative errors using the DQZ (N=9) and DQU (N=7) approaches are 0.2% and 0.02%, respectively, but it is also noted that the DQZ (N=8), DQN (N=8) and DQU (N=6) solutions for this case is 0.25%, 0.27% and 0.5%. The DQZ (N=9) and DQU(N=7) solutions for static deflection of SS-SS-SS-SS plate are 0.05% and 0.1%, respectively. Therefore, it is concluded that the DQU technique may be very efficient only under some specific situations.

In section 2.2 we discuss the formulation-H for geometrically nonlinear vibration of beams. In the following we continue this work. The DQN results by Feng and Bert (1992) for the clamped-clamped case agreed very well with existing FEM solutions by Mei (1973). However, their results for the simply support-simply supported case showed somewhat large discrepancy with FEM (Mei, 1973) and analytical solutions (Singh and Rao, 1990). This is because the conventional DQN approach was not very successful for SS-SS boundary conditions (Wang and Bert, 1993). In the following analysis, we will show Wang and Bert's DQU approach to be very efficient for the analysis of nonlinear structural components with SS-SS boundary conditions (Chen et al., 1996j).

It is known that, in Wang and Bert's new approach, there exists

$$\overline{D} = \overline{B}^2 \tag{4.4-10}$$

for SS-SS boundary conditions. Therefore, $\overline{B}$ and $\overline{D}$ are orthogonal similarity and both have the same eigenvectors. So the beam oscillates at the same mode as the one in the linear case, and the iterative procedures for the solution of equation (2.2-7) used in Feng and Bert (1992) are not necessary for the SS-SS case. In this study, we first solve the eigenvalue and eigenvectors of $\overline{B}$ and $\overline{D}$, and then obtain the nonlinear coefficient of equation (2.2-7) by using these eigenvectors. The resulting dimensionless nonlinear frequency can be obtained by

$$\varpi = \sqrt{\lambda_{\overline{D}} - \eta\lambda_{\overline{B}}} \tag{4.4-11}$$

where $\lambda_{\overline{B}}$ and $\lambda_{\overline{D}}$ are the eigenvalues of $\overline{B}$ and $\overline{D}$, η is nonlinear coefficient. If choosing the minimum values of $\lambda_{\overline{B}}$ and $\lambda_{\overline{D}}$, the fundamental dimensionless nonlinear frequency is obtained.

Seven equally spaced grid points are used in the present DQ computation. Table 4-3 shows the remarkable agreement between the analytical, finite element and present DQ solutions. Amplitude-frequency curves are plotted in Figure 4-1. Obviously, DQU gives more accurate results than the conventional DQN by Feng and Bert (1992). As is expected, the DQ solutions using the zeros of the Chebyshev polynomial of seven order are more accurate than using equally spaced grid points, and in this case are coincident with the analytical ones. Compared with FEM, the DQ method yields more exact results and is easier to be used and requires much less computational effort and storage.

Circular plates with clamped or simply supported edge under uniformly distributed loading have been studied by Bert et al. (1988, 1994), Jang et al. (1989), Chen et al. (1993), Wang (1995) by using the DQN, DQU and DQZ approaches, respectively. We only compare the numerical results here. For the details see the related references. The DQU and DQZ approaches can give analytical solutions for static bending of circular plates, while the DQN approach can not. As was pointed out by Bert and Malik (1996d), the DQU technique has some limitations in practical engineering. Therefore, it is our conclusion that the DQZ method is in general preferred in the DQ solution of problems with multiple boundary conditions. Also, the DQN, DQU and DQZ solutions of geometrically nonlinear bending of plates are discussed in later section 4.5. We can obtain the same conclusion as in linear cases.

Table 4-3. The ratios (ω_n/ω_l) of the nonlinear frequencies to the linear frequency for a SS-SS beam.

a/r	Elliptical integral[6]	Analytical	Present DQ	DQ[1]	FEM[2]
0.1	1.0009	1.0009	1.0009	1.0010	1.0009
0.2	1.0037	1.0037	1.0037	1.0043	1.0037
0.4	1.0149	1.0149	1.0149	1.0170	1.0148
0.6	1.0331	1.0332	1.0332	1.0384	1.0329
0.8	1.0580	1.0583	1.0582	1.0673	1.0578
1.0	1.0892	1.0897	1.0896	1.1030	1.0889
1.5	1.1902	1.1924	1.1922	1.2045	1.1902
2.0	1.3178	1.3229	1.3225	1.3170	1.3183
3.0	1.6257	1.6394	1.6389	–	1.6260
4.0	1.9760	2.0000	1.9991	–	1.9715
5.0	2.3501	2.3848	2.3836	–	2.3341

Results underlined are incorrectly typed in Feng and Bert (1992).

What follows may be outside our scope of the title of this section. A straightforward and intuitive procedure is given to obtain the analytical solution of governing equation of geometrically nonlinear vibration of SS-SS beam.

The m order normal mode of a linear SS-SS beam is

$$v(\xi) = \sin(m\pi\xi). \tag{4.4-12}$$

Based on the fact that the nonlinear SS-SS beam has the same vibrational mode as the linear SS-SS beam, we obtain nonlinear frequency of m order mode for geometrically nonlinear SS-SS beam by substituting equation (4.4-12) into equation (2.2-7)

$$\varpi = (m\pi)^2\sqrt{1 + \frac{3}{16}\frac{a^2}{r^2}}. \tag{4.4-13}$$

The above solution is coincident with that by using the perturbation method (Singh et al., 1990) and is regarded as the analytical solution of the m order mode. Thus,

$$\frac{\varpi}{\omega_l} = \sqrt{1 + \frac{3}{16}\frac{a^2}{r^2}}, \tag{4.4-14}$$

where ω_l =(mπ)^2 is the linear frequency of m order model.

The elliptic integral solutions for assumed space model (ASM) in this case by Woinowsky-Krieger (1950) are also listed in table 4-3 and compared with the analytical solutions of the governing equation (2.2-7) in Figure 4-2. It is noted that both agree well especially when a/r is less than 2.0. Therefore, it is concluded that the governing equation (2.2-7), e.g. the so-called assumed time model (ATM), provides a rather accurate description for the geometrically nonlinear vibration of SS-SS beam.

4.5. Geometrically Nonlinear Bending Analysis of Plates

The geometrically nonlinear behavior of thin plates is usually described by the von Karman equations and has become a benchmark problem for testing numerical solutions to nonlinear partial differential equations (Timoshenko and Woinowsky-Krieger, 1959). Bert et al. (1989) have used the DQ method to solve the static von Karman equations in analyzing geometrically nonlinear bending of isotropic and orthotropic rectangular plates. In the present study, we wish to apply some new techniques, presented in the foregoing sections, to simplify the use and improve efficiency and accuracy in applying DQ method for these cases. The main purpose of this section is to show the utility, simplicity and high efficiency of the Hadamard product and SJT product approach as well as new matrix approximate formulas in the DQ nonlinear computations (Chen et al., 1996h).

Considering a thin, homogeneous, orthotropic rectangular plate subject to a uniformly distributed transverse load (Bert et al. 1989), we have

$$E_1 u_{,xx} + \mu G_{12} u_{,yy} + C v_{,xy} = -w_{,x}\left(E_1 w_{,xx} + \mu G_{12} w_{,yy}\right) - C w_{,y} w_{,xy} \tag{4.5-1a}$$

$$E_2 v_{,yy} + \mu G_{12} v_{,xx} + C u_{,xy} = -w_{,y}\left(E_2 w_{,yy} + \mu G_{12} w_{,xx}\right) - C w_{,x} w_{,xy} \tag{4.5-1b}$$

$$\begin{aligned} D_1 w_{,xxxx} + 2D_3 w_{,xxyy} + D_2 w_{,yyyy} &= q + \frac{h}{\mu}\left[\left(u_{,x} + \frac{1}{2}w_{,x}^2\right)\left(E_1 w_{,xx} + \upsilon_{12} E_2 w_{,yy}\right)\right. \\ &\left. + \left(v_{,y} + \frac{1}{2}w_{,y}^2\right)\left(E_2 w_{,yy} + \upsilon_{21} E_1 w_{,xx}\right) + 2\mu G_{12} w_{,xy}\left(u_{,y} + v_{,x} + w_{,x} w_{,y}\right)\right] \end{aligned} \tag{4.5-1c}$$

in terms of three displacement components, where ν_{12} and ν_{21} are Poisson's ratio, E_1 and E_2 are the Young's moduli. C is the shear modulus, D_1, D_2 and D_4 are the principal bending and twisting rigidities, $\mu=1-\nu_{12}\nu_{21}$, u, v and w are the desired inplane and transverse displacements. a, b and h are width, length and thickness of plate, respectively. The equations are high nonlinearity, which consists of two second order and one fourth order simultaneous cubic nonlinear partial differential equations.

Applying the Hadamard product and the presented DQ matrix approximate formulas (2-7), the formulations for this case are

$$\begin{aligned} E_1 \overline{B}_x U + \mu G_{12} U \overline{B}_y^T + C \overline{A}_x V \overline{A}_y^T &= -\left(\overline{A}_x W\right) \circ \left(E_1 \overline{B}_x W + \mu G_{12} W \overline{B}_y^T\right) \\ &- C\left(W \overline{A}_y^T\right) \circ \left(\overline{A}_x W \overline{A}_y^T\right) \end{aligned} \tag{4.5-2a}$$

$$\begin{aligned} E_2 V \overline{B}_y^T + \mu G_{12} \overline{B}_x V + C \overline{A}_x U \overline{A}_y^T &= -\left(W \overline{A}_y^T\right) \circ \left(E_2 W \overline{B}_y^T + \mu G_{12} \overline{B}_x W\right) \\ &- C\left(\overline{A}_x W\right) \circ \left(\overline{A}_x W \overline{A}_y^T\right) \end{aligned} \tag{4.5-2b}$$

$$
\begin{aligned}
D_1\overline{D}_x W + 2D_3\overline{B}_x W\overline{B}_y^T + D_2 W\overline{D}_y^T = q + \frac{h}{\mu}\Bigg[\bigg[\overline{A}_x U + \frac{1}{2}\left(\overline{A}_x W\right)^{\circ 2}\bigg] \\
\circ\left[E_1\overline{B}_x W + \upsilon_{12}E_2 W\overline{B}_y^T\right] + \left[V\overline{A}_y^T + \frac{1}{2}\left(W\overline{A}_y^T\right)^{\circ 2}\right]\circ\left[E_2 W\overline{B}_y^T + \upsilon_{21}E_1\overline{B}_x W\right] \\
+ 2\mu G_{12}\left(\overline{A}_x W\overline{A}_y^T\right)\circ\left[U\overline{A}_y^T + \overline{A}_x V + \left(\overline{A}_x W\right)\circ\left(W\overline{A}_y^T\right)\right]\Bigg]
\end{aligned}
\tag{4.5-2c}
$$

where $\overline{A}$, $\overline{B}$ and $\overline{D}$ with subscript x and y denote the modified weighting coefficient matrix along x and y directions, respectively. The orders of these matrices are N-2 for the DQU approach and N-4 for the DQZ and DQN approaches, where N is the number of grid points. Note that the boundary conditions have applied in the DQU, DQZ and DQN approaches and, thus, are no longer considered. U, V and W in the above formulations are rectangular matrices. Also, $\vec{A}_x$ and $\vec{B}_x$ along the x-direction and $\vec{A}_y$ and $\vec{B}_y$ along y-direction are, respectively, the same for the desired displacement U, V and W in the cases of clamped and simply supported edges.

By using the Kronecker product of matrices, we have

$$H_1\vec{U} + H_2\vec{V} = -\left(H_7\vec{W}\right)\circ\left(H_1\vec{W}\right) - \left(H_8\vec{W}\right)\circ\left(H_2\vec{W}\right) \tag{4.5-3a}$$

$$H_2\vec{U} + H_3\vec{V} = -\left(H_8\vec{W}\right)\circ\left(H_3\vec{W}\right) - \left(H_7\vec{W}\right)\circ\left(H_2\vec{W}\right) \tag{4.5-3b}$$

$$
\begin{aligned}
H_4\vec{W} = \frac{qa^4}{D_1 h} + \frac{a^4}{\mu D_1 h}\Bigg[\frac{a^2}{h^2}\left[H_7\vec{U} + \frac{1}{2}\left(H_7\vec{W}\right)^{\circ 2}\right]\circ\left[H_5\vec{W}\right] + \frac{b^2}{h^2}\left[H_8\vec{V} + \frac{1}{2}\left(H_8\vec{W}\right)^{\circ 2}\right] \\
\circ\left[H_6\vec{W}\right] + \frac{2\mu G_{12}}{C}\left(H_2\vec{W}\right)\circ\left[H_8\vec{U} + H_7\vec{V} + \left(H_7\vec{W}\right)\circ\left(H_8\vec{W}\right)\right]\Bigg]
\end{aligned}
\tag{4.5-3c}
$$

where $\vec{W}$, $\vec{U}$ and $\vec{V}$ are vectors yielded by stacking the rows of the corresponding rectangular matrix W, U, and V into one long vector. H_1, H_2, H_3, H_4, H_5, H_6, H_7 and H_8 are defined as follows:

$$H_1 = E_1\left(\overline{B}_x \otimes I_y\right) + \mu G_{12}\left(\frac{a}{b}\right)^2\left(I_x \otimes \overline{B}_y\right) \tag{4.5-4a}$$

$$H_2 = C\left(\overline{A}_x \otimes \overline{A}_y\right) \tag{4.5-4b}$$

$$H_3 = E_2\left(I_x \otimes \overline{B}_y\right) + \mu G_{12}\left(\frac{b}{a}\right)^2\left(\overline{B}_x \otimes I_y\right) \tag{4.5-4c}$$

$$H_4 = \overline{D}_x \otimes I_y + \frac{D_2}{D_1}\left(\frac{a}{b}\right)^2\left(\overline{B}_x \otimes \overline{B}_y\right) + \frac{D_3}{D_1}\left(\frac{a}{b}\right)^4\left(I_x \otimes \overline{D}_y\right) \tag{4.5-4d}$$

$$H_5 = E_1\left(\frac{h}{a}\right)^2\left(\overline{B}_x \otimes I_y\right) + \upsilon_{12}E_2\left(\frac{h}{b}\right)^2\left(I_x \otimes \overline{B}_y\right) \tag{4.5-4e}$$

$$H_6 = E_2\left(\frac{h}{b}\right)^2\left(I_x \otimes \overline{B}_y\right) + \upsilon_{21}E_1\left(\frac{h}{a}\right)^2\left(\overline{B}_x \otimes I_y\right) \tag{4.5-4f}$$

$$H_7 = \frac{h^2}{a^2}\left(\overline{A}_x \otimes I_y\right) \tag{4.5-4g}$$

$$H_8 = \frac{h^2}{b^2}\left(I_x \otimes \overline{A}_y\right) \tag{4.5-4h}$$

Obviously, the nonlinear formulations are very easily accomplished by using the new matrix approximate formulas and Hadamard product. The present matrix form is also much simpler and more explicit than the conventional algebraic polynomial form given by Bert et al. (1989).

The variables are nondimensionalized as $X \equiv x/a, Y \equiv y/b, U \equiv u/a, V \equiv v/b$ and $W \equiv w/h$. Equations (4-3a) and (4-3b) can be also restated as

$$H_1\vec{U} + H_2\vec{V} = -L_1(\vec{W}) \tag{4.5-5a}$$

$$H_2\vec{U} + H_3\vec{V} = -L_2(\vec{W}) \tag{4.5-5b}$$

where

$$L_1(\vec{W}) = (H_7\vec{W}) \circ (H_1\vec{W}) + (H_8\vec{W}) \circ (H_2\vec{W}) \tag{4.5-6a}$$

$$L_2(\vec{W}) = (H_8\vec{W}) \circ (H_3\vec{W}) + (H_7\vec{W}) \circ (H_2\vec{W}) \tag{4.5-6b}$$

The unknown vector $\vec{U}$ and $\vec{V}$ can be expressed in terms of $\vec{W}$ by

$$\vec{U} = H_9^{-1}H_3^{-1}L_2(\vec{W}) - H_9^{-1}H_2^{-1}L_1(\vec{W}) \tag{4.5-7a}$$

and

$$\vec{V} = H_{10}^{-1}H_2^{-1}L_2(\vec{W}) - H_{10}^{-1}H_1^{-1}L_1(\vec{W}) \tag{4.5-7b}$$

where $H_9 = H_2^{-1}H_1 - H_3^{-1}H_2$ and $H_{10} = H_1^{-1}H_2 - H_2^{-1}H_3$. By using the SJT product for the evaluation of the Jacobian matrix, we have

$$\frac{\partial\vec{U}}{\partial\vec{W}} = H_9^{-1}H_3^{-1}\frac{\partial L_2(\vec{W})}{\partial\vec{W}} - H_9^{-1}H_2^{-1}\frac{\partial L_1(\vec{W})}{\partial\vec{W}} \tag{4.5-8a}$$

and

$$\frac{\partial\vec{V}}{\partial\vec{W}} = H_{10}^{-1}H_2^{-1}\frac{\partial L_2(\vec{W})}{\partial\vec{W}} - H_{10}^{-1}H_1^{-1}\frac{\partial L_1(\vec{W})}{\partial\vec{W}} \tag{4.5-8b}$$

where

$$\frac{\partial L_1(\vec{W})}{\partial\vec{W}} = H_7\Diamond(H_1\vec{W}) + H_1\Diamond(H_7\vec{W}) + H_8\Diamond(H_2\vec{W}) + H_2\Diamond(H_8\vec{W}) \tag{4.5-9a}$$

and

$$\frac{\partial L_2(\vec{W})}{\partial\vec{W}} = H_8\Diamond(H_3\vec{W}) + H_3\Diamond(H_8\vec{W}) + H_7\Diamond(H_2\vec{W}) + H_2\Diamond(H_7\vec{W}). \tag{4.5-9b}$$

$\frac{\partial\vec{U}}{\partial\vec{W}}$ and $\frac{\partial\vec{V}}{\partial\vec{W}}$ are relative Jacobian derivative matrices of dependent variable vector $\vec{U}$ and $\vec{V}$ with respect to $\vec{W}$. By applying formulas (4.5-7a, b) and (4.5-8a, b), the coupling formulation equations (4.5-3a, b, c) are decoupled. The resulting set of simultaneous nonlinear algebraic equations are reduced from 3(N-2)×3(N-2) to (N-2)×(N-2) for the plate with simply supported edges or from 3(N-4)×3(N-4) to (N-4)×(N-4) for the plate with clamped edges. It is known that each iteration step in the Newton-Raphson method is necessary to solve a set of linear simultaneous algebraic equations and requires an order of n^3 scalar multiplications, where n is the size of the equations. For example, the Gauss elimination method requires $n^3/3$ scalar multiplications. Therefore, the computational effort and storage requirements here are only about one twenty-seventh and one-ninth, respectively, as much as in Bert et al. (1989). $\vec{W}$ is a basic variable vector here. Equation (4.5-3c) is chosen as the basic iteration equation, namely,

$$\begin{aligned}\varphi\left\{\vec{W}\right\} = H_4\vec{W} - \frac{a^4}{\mu D_1 h}&\left[\frac{a^2}{h^2}\left[H_7\vec{U} + \frac{1}{2}\left(H_7\vec{W}\right)^{\circ 2}\right]\circ\left[H_5\vec{W}\right] + \frac{b^2}{h^2}\left[H_8\vec{V} + \frac{1}{2}\left(H_8\vec{W}\right)^{\circ 2}\right]\right. \\ &\left.\circ\left[H_6\vec{W}\right] + \frac{2\mu G_{12}}{C}\left(H_2\vec{W}\right)\circ\left[H_8\vec{U} + H_7\vec{V} + \left(H_7\vec{W}\right)\circ\left(H_8\vec{W}\right)\right]\right] - \frac{qa^4}{D_1 h} = 0\end{aligned} . \quad (4.5\text{-}10)$$

The Jacobian derivative matrix for the above iteration equation is given by

$$\begin{aligned}\frac{\partial\varphi\left\{\vec{W}\right\}}{\partial\vec{W}} = H_4 - \frac{a^4}{\mu D_1 h}&\left[\frac{a^2}{h^2}\left[H_7\frac{\partial\vec{U}}{\partial\vec{W}} + H_7\Diamond\left(H_7\vec{W}\right)\right]\Diamond\left[H_5\vec{W}\right]\right. \\ &+\frac{a^2}{h^2}H_5\Diamond\left[H_7\vec{U} + \frac{1}{2}\left(H_7\vec{W}\right)^{\circ 2}\right] + \frac{b^2}{h^2}\left[H_8\frac{\partial\vec{V}}{\partial\vec{W}} + H_8\Diamond\left(H_8\vec{W}\right)\right]\Diamond\left[H_6\vec{W}\right] \\ &+\frac{b^2}{h^2}H_6\Diamond\left[H_8\vec{U} + \frac{1}{2}\left(H_8\vec{W}\right)^{\circ 2}\right] + \frac{2\mu G_{12}}{C}H_2\Diamond\left[H_8\vec{U} + H_7\vec{V} + \left(H_7\vec{W}\right)\circ\left(H_8\vec{W}\right)\right] \\ &\left.+\frac{2\mu G_{12}}{C}\left[H_8\frac{\partial\vec{U}}{\partial\vec{W}} + H_7\frac{\partial\vec{V}}{\partial\vec{W}} + H_7\Diamond\left(H_8\vec{W}\right) + H_8\Diamond\left(H_7\vec{W}\right)\right]\Diamond\left(H_2\vec{W}\right)\right]\end{aligned} . \quad (4.5\text{-}11)$$

It is noted that the SJT product approach here yields the analytical solution of the Jacobian matrix quite simply and efficiently. The Newton-Raphson iteration equation for this case is the same as equation (2.5-7).

The solutions obtained by the DQ method for the corresponding linear isotropic and orthotropic plates are chosen as the initial guess of the iteration procedures. Even if the resulting nonlinear results are even eight times larger than the initial linear solutions, the Newton-Raphson method still converges. Therefore, the Newton-Raphson method has rather big convergence domain for these cases. Moreover, the iterative times varies from 1 to 7 for various loading and the solutions converge very rapidly. In contrast, the IMSL subroutine NEONE used in the Bert et al. (1989) computed the Jacobian matrix approximately by a finite difference technique. Therefore, the accuracy and convergence speed were affected.

It is well known that the accuracy and stability of the DQ method can be improved significantly if the Chebyshev grid spacing is used. In the following the DQU and DQZ solutions are obtained by using Chebyshev grid spacings 7×7 for a simply supported plate and 11×11 for a clamped plate unless where specified. To avoid the effects of round-off errors on the accuracy of the solution, double-precision arithmetic is used in all the analysis presented here. Bert et al. (1989) has pointed out the high efficiency and ease of use in the DQ method in comparison to other numerical techniques such as the finite element, finite difference, perturbation, Galerkin and Rayleigh-Ritz, etc., while this study places its emphasis in the simplification of the use and further improvement of the accuracy and efficiency in the DQ method. Therefore, the numerical comparisons with other numerical techniques are not repeated here.

The same simply-supported and clamped isotropic square plates subject to a uniformly distributed loading as in example 1 of Bert et al. (1989) are recalculated by the present DQ method. The results are shown in Fig. 4-3, and compared with the exact Levy (1942a, b) and the conventional DQ (Bert et al., 1989) solutions. The present DQU and DQZ results all show remarkable agreement with those of Levy (1942a) and Yang (972). It is also noted that the DQN approach using 7×7 grid points by Bert et al. (1989) gives obviously better results in the clamped cases than in the simply supported cases. This is because the DQN approach is not suitable for the cases of supported edges. As is expected, the DQU approach gives more accurate results than the DQN approach for simply supported plate. Therefore, the former is a competitive alternative to the latter for the nonlinear cases of simple supports. Both the DQZ and DQN approaches can yield accurate solutions for the clamped plate. But it should be emphasized that the DQN approach can not use larger number of grid points as in the DQZ approach due to instability caused by the δ effect.

Also, we compute the isotropic simply supported square plate under a uniformly distributed transverse load. The parameters of this case are a=16”, h=0.1”, E=30E+6 and ν=0.316. Fig. 4-4 depicts the results obtained by the DQU approach using 5×5 Chebyshev grids and 7×7 equally spaced grids. All solutions agree very well with those given by Levy (1942a). The DQ method is demonstrated again to be highly computationally efficient for nonlinear structural analysis. As is expected, the DQU method using the Chebyshev points yields more accurate results than using equally spaced grids, although in this case more equally grid points are used.

The central deflection of the clamped plate (a=100, h=1.0, E=2.1E+6, ν=0.316, q=3.0) and the simply-supported square plate (a=100, h=1.0, E=2.1E+6, ν=0.25, q=1.0) subject to a uniformly distributed pressure are computed by the DQU method and listed in table 4-4. The 9×9 Chebyshev grid spacing is used for the case of clamped edges. The present DQU solutions show excellent agreement with the analytical (Bazeley et al., 1965) and FEM solutions (Bazeley et al., 1965; Zhu et al., 1989). However, the computational effort in the present DQ method is much less than in the analytical method and FEM.

Table 4-4. The Central deflections of the clamped and simply-supported square plates

Methods	Analytical	Bazeley	Zhu	Present
Simply supported	0.940	1.028(9.3%)	0.942(0.3%)	0.944(0.4%)
Clamped	1.151	1.316(14.3%)	1.170(1.6%)	1.123(2.4%)

The numerical examples on the orthotropic square plate provided by Bert et al. (1989) are recalculated by the present DQ methods. The specific parameters are E_1=18.7E+6 psi; E_2=1.3E+6 psi, G_{12}=0.6E+6 psi; ν_{12}=0.3; h=0.0624inch; a=9.4; b=7.75inch. The center deflections for the clamped and simply supported cases are displayed in Figures 4-5 and 4-6, respectively. For the case of clamped edges, the results by the DQZ method using 11×11 grid spacing are very close to those by Bert and Cho (1988b). It is noted that the DQZ method using 15×15 or even 21×21 grid spacings is still stable and give accurate results, but computational effort also increases exponentially. For the case of simple supports, it is noted that the results using DQN approach under 7×7 grid spacing given by Bert et al. (1989) are obviously larger than those by Bert and Cho (1988a). In contrast, the present DQU approach appears to give results that are much closer to those by Bert and Cho (1988b) as shown in Fig. 4-6. Bert and Cho’s (1988b) values are taken from graphs 9 and 10 in Bert et al. (1989) with appropriate scaling factors.

The above computations were executed on an IBM-PC 386DX computer with 4M memory. Microsoft Fortran77 ver. 3.3 and NDP-Fortran-386 ver. 2.1 are used for programming. It is found that the results yielded by using both Fortran versions have slight discrepancy, as an example, for the case of a simply supported plate depicted in Fig. 4-7, the nondimensionalized center deflection under q=2.5p.s.i. are 0.977 for Microsoft Fortran77 ver. 3.3 and 0.974 for NDP-Fortran-386 ver. 2.1, respectively. The results shown in this paper are all obtained by using NDP-Fortran-386 ver. 2.1.

The DQ approach using some new techniques is applied to analyze geometrically nonlinear bending of isotropic and orthotropic plates with simply supported and clamped edges. It is apparent that the results obtained are more accurate than the conventional ones (Bert et al., 1989) and compare favorably with exact solutions (Levy, 1942a, b; Zhu et al., 1989). The DQU approach is proved to be a successful technique for geometrically nonlinear plate with SS-SS boundary conditions. The DQZ approach is improved and shown to be a stable and accurate technique for handling the cases with the C-C boundary conditions. Although only square simply supported and clamped boundary conditions are involved in this study, it is straightforward that the same procedures can be easily used for handling problems with various aspect ratios and other edge conditions.

The references in which the DQ method was applied to deal with nonlinear problems are still few due to much more complex programming, storage requirements, formulation and computing effort in comparison to linear problems. The Hadamard product and SJT product approach may provide a simple and efficient technique to greatly reduce the above difficulties. The detailed solution procedures are provided here to show the simplicity and efficiency of the Hadamard and SJT product approach as well as new DQ matrix approximate formulas. For more complex plates with varying thickness, Poisson's ratio and Young's modulus, the DQ formulation and the evaluation of the respective Jacobian matrix are also easily finished by using the Hadamard product and SJT product. The extension of the present DQ method to the nonlinear dynamic and postbuckling cases is currently the subject of further investigation. Also, it is expected that the Hadamard product and SJT product techniques can achieve a considerable savings in formulation, storage and CPU time for other nonlinear examples such as those discussed by Striz et al. (1988, 1994a), Shu et al. (1992a, b, 1994a, b) and Wang (1994b).

Chapter 5.

INITIAL VALUE PROBLEMS

5.1. Introduction

The numerical solution of initial value problems has been of vital importance in many science and engineering areas. Considerable research effort has been devoted to the development of efficient computational method for the solution of the ordinary equations of initial value problems. So far there are several numerical methods available such as the known Houbolt, Wilson θ, Newmark, central difference, Runge-Kutta and Gear, etc. (Dokainish and Subbaraj, 1989a, b). The purpose of this chapter is to apply the differential quadrature method in using some new techniques to initial value problems. The conventional DQ method (Civan and Sliepcevich, 1984a) is to convert a very simple initial value problems into a boundary value problem by means of an algebraic transformation. Therefore, it is in general not significant in practice. It is worth pointing out that our work is the first authentic attempt to solve initial value problem using computational step-by-step procedure of the DQ method.

In the literature, the DQ method has been usually applied to handle boundary value problems. There is an impediment to using the DQ method for the initial value problems. It is when the DQ method using m grid points is applied to a system of N ordinary differential equations, a system of N×m simultaneous algebraic equations will have to be solved and, thus, is costly in comparison to other existing methods.

In this study, we introduce the DQ approximate formulas in matrix form for ordinary differential equations of initial value problems. By using these approximate formulas, the DQ formulation for initial problems can be obtained very easily and is a Lyapunov-like algebraic matrix equation. The reduction techniques discussed in section 4.2 for solving the Lyapunov algebraic matrix equation can be successfully applied to reduce the computational effort greatly in the solution of the present formulation. Wang and Bert’s DQU technique is extended to the DQ formulation of the initial value problems and proves to be accurate and efficient. Based on the fact that the DQ method is equivalent to the collocation method, we point out that the DQ method is an A-stable method. Thus, the DQ method is safe for structural dynamic analysis and stiff problems. According to the new formulas of the truncation error presented in section 3.2, it is apparent that the DQ method for the initial value problems is high order accuracy, namely, the convergence speed is O(τ^{N-2}). We also analyze advantages of the DQ method over the collocation method. Some numerical examples are also provided to demonstrate the efficiency, reliability and simplicity of the DQ analysis of structural dynamic and stiff problems using the above-mentioned new techniques. Finally, we also discuss some promsing approaches to overcome the difficulty in handling nonlinear initial value problems.

5.2. Approximate Formulas in Matrix Form

The DQ weighting coefficient matrix for the first and second derivatives are denoted as A and B in this section, respectively. Considering the ordinary differential equations

$$\frac{d\bar{u}}{dt} = H\bar{u} + c \tag{5.2-1}$$

where $\bar{u}$ is n×1 desired vector, H is a constant matrix, c is a n×1 constant vector, t is a scalar variable. By analogy with matrix approximate formulas (4.2-1) for boundary value problems, the formulation for equation (5.2-1) is taken to be of the matrix form

$$U\bar{A}^T = HU + C \tag{5.2-2}$$

where $\overline{A}$ is the DQ weighting coefficient matrix, modified by the initial condition, for the 1st order derivative of unknown function u with respect to variable t. Capital letter U and C denote n×m rectangular matrix stacked from vector $\vec{u}$ and c in Eq. (5.2-1) along variable t direction, namely,

$$U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1m} \\ u_{21} & u_{22} & \cdots & u_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n1} & u_{n2} & \cdots & u_{nm} \end{bmatrix} \tag{5.2-3}$$

and

$$C = \begin{bmatrix} c_1 & c_1 & \cdots & c_1 \\ c_2 & c_2 & \cdots & c_2 \\ \vdots & \vdots & \ddots & \vdots \\ c_n & c_n & \cdots & c_n \end{bmatrix} \tag{5.2-4}$$

where m is the number of grid points along variable t direction. Matrix formulations (5.2-2) for ordinary differential equations is a key formula in the present study. It is emphasized that the formulation (5.2-1) is effective only when the initial conditions for all unknown elements u_j in vector $\vec{u}$ are the same and, thus, all corresponding weighting coefficient matrices $\overline{A}$ are identical. This requirement can be satisfied easily by using simple algebraic transformation, which will be given in the latter analysis. Obviously, there exists similar matrix formulation for higher ordinary differential equations of initial value problems. Also, it is noted that formulation equation (5.2-2) is a Lyapunov-like algebraic matrix equation.

As can be seen from table 4-I, the R-THR method (Gui, 1992) may be the most efficient in the solution of steady-state Lyapunov matrix equations. However, it is noted that the present problems is involved transient initial value equation, and thus a different estimation on total computational effort is required. Step 1 in the methods, which costs major calculation time as shown in table 4-1, is required to operate only once for equations (5.2-2). Therefore, the computation effort in step 1 takes a very small percentage of total solution time and is in general neglected in a long time journey. Linear transformations are also operated only once and thus can be neglected. The actual computing cost is to accumulate computational effort in step 2, 3 and 4. Therefore, the BS method may be the most efficient method for initial value problems. The equation (5.2-2) is used as a sample example in the following discussion. The total multiplication times in the BS method are $2.5m^2n+2.5mn^2$ for equation (5.2-2) in each time step. Since n (the dimension number of equation (5.2-1)) is usually much larger than m (the number of inner grid points) in practical engineering. So the computing effort can be estimated approximately as $2.5mn^2$ in each step or $2.5n^2$ at each time grid point, which is basically the same as in the Wilson θ, Houbolt and Newmark methods (Bathe and Wilson, 1976). It is still possible to apply the sparseness and symmetricity of matrices G and R in equation (5.2-2) to further reduce the computational effort. It should be also emphasized that the BS, HS and R-THR methods have very high efficiency for the parallel computations (Gui, 1992). The numerical examples for the DQ analysis of initial value problems are presented in sections 5.3, 5.4 and 5.5.

5.3. Stability Aspect and Comparison With the Collocation Method

Bellman et al. (1972) pointed out that the stability analysis was very difficulty problems in the DQ method, especially for the nonlinear problems. The study on the stability for the DQ method is not available in the existing literature. Section 2.9.3 discussed nonlinear stability analysis in the DQ method for the steady-state problems by means of the Hadamard product (Chen et al., 1996f). The present emphasis is placed in the stability aspect of the DQ solution of the initial value problems.

Quan et al. (1989a), Bert et al. (1993) and Mansell et al. (1993) have pointed out that the DQ method is in fact equivalent to the general collocation (pseudo-spectral) method. It is known that the collocation method is an implicit A-stable method (Burka, 1982). Therefore, the DQ method is also an A-stable method, namely for $\mathrm{Re}(\lambda_i)\angle 0$, there are no limits on the size of the product $|\tau\lambda|$, where λ_i are the eigenvalues of the coefficient matrix of differential equations and τ is the step size of variable t. Since the real part of eigenvalues in structural dynamic equation is in general negative, an A-stable method is also unconditionally stable for structural dynamic analysis. It is well known that A-stable method is reliable for stiff problems. Since the collocation method has been applied to compute the initial value problems (Burka, 1982; Wright, 1964; Finlayson, 1972; Villdsen, 1972). In the following we will point out the advantages of the DQ method for the same task comparatively.

The salient merits of the DQ method is its ease of implementation (Mansell et al., 1993). The DQ method directly computes the unknown function values at grid points rather than the spectral variables as in the collocation method. Moreover, these desired spectral coefficients usually have no physical significance and thus assumed initial values in the solution of the nonlinear problems are inherently poor guesses. So the computational effort for the nonlinear problems is aggravated (Burka, 1982). This shortcoming does not exist in the DQ method. Next, for the definite basis functions, the DQ weighting coefficients for certain grid spacing need be computed only once and are independent of any special problems. Therefore, these weighting coefficients can be used repeatedly for various problems. Also, as was pointed out by Quan and Chang (1989b), the DQ method using the zeros of the Chebyshev polynomials has somewhat faster convergence rate than the orthogonal collocation method. The DQ method is also more convenient to use any grid spacing.

In addition, the most important point may be that the presented Lyapunov-like matrix formulation (5.2-2) in the DQ method drastically reduces formulation and computing effort and storage requirements for the initial value problems. For N linear ordinary differential equations, the collocation method using m grid points yields a system of m×N linear algebraic equations at each iteration step. If they are to be solved simultaneously, the repeated computation of m×N matrices for linear problems and even larger matrices for nonlinear problems are too costly (Burka, 1982). This is main obstacle to apply the collocation method for initial value problems. As discussed in section 5.2, the computing effort in the DQ method using the Lyapunov-like matrix formulation is reduced to nearly the same as in the conventional single-step numerical integration methods such as Wilson θ and Newmark, etc. Based on the above considerations, it is believed that the DQ method is preferred for numerical computations of the present purpose.

5.4. Structural Dynamic Problems

Typical structural dynamic problems have natural periods that differ by many orders of magnitude. Thus, they are in general “stiff”. The methods for solving problems of such type are required to be A-stable, that is, unconditionally stable. The often used Newmark, Wilson θ, and Houblot methods is also unconditionally stable. However, the maximum order of an A-stable multistep method was 2 (Dahlquist, 1963). Therefore, the computing efficiency of these methods is not high. On the other hand, the explicit methods such as central difference and Runge-Kutta methods are conditionally stable, and, thus, require excessively small time steps to ensure numerical stability. Consequently, the computing cost is unnecessarily high, and is generally not practical. This section aims to apply the DQ method to structural dynamic problems (Chen et al., 1996g). To the authors' knowledge, the method has been not used for solving such problems in the existing literature.

The linear matrix equation of structural dynamic for general purpose is

$$Mu'' + Cu' + Ku = f\ , \tag{5.4-1}$$

where M, C and K are time-invariant mass, damping, and stiffness matrices, respectively. f is the vector of applied loads, u is the displacement vector and superposed dotes indicate time differentiation, i.e.,$u'' = d^2u/dt^2$, $u' = du/dt$. The initial conditions for the problem are

$$u\big|_{t=t_1} = d \tag{5.4-2}$$

$$u'\big|_{t=t_1} = h \tag{5.4-3}$$

where d and h are the prescribed vectors of initial data, t_1 is the initial time. The order of the above dynamic equation is assumed as n.

The problems is usually complicated if considering the influence of damping. The assumption of proportional damping is in general adequate in many engineering analysis. However, in the analysis of structures with widely varying material properties, nonproportional damping may be needed (Dokainish and Subbaraj, 1989). In the following we derive the DQ formulations for the problems with nonproportional and proportional damping, respectively. The resulting form of two formulations is also a Lyapunov algebraic matrix equation.

5.4.1. Formulation for the Problems with Nonproportional Damping

We here discuss general cases, in which the definition of damping is very broad and include non-proportional. Using the transformation presented by Zhong (1995), Eq. (4-5) can be restated as

$$v' = Hv + q \tag{5.4-4}$$

in which

$$v = \{u, p\}^T, \qquad p = Mu' + Cu/2\,, \qquad H = \begin{bmatrix} D & G \\ E & F \end{bmatrix},$$

$$D = -M^{-1}C/2, \qquad E = CM^{-1}C/4 - K, \qquad F = -CM^{-1}/2\,,$$
$$G = M^{-1}, \qquad q = \{0.f\}^T \tag{5.4-5}$$

The corresponding initial conditions are

$$v\big|_{t=t_1} = v_0 = \{d, p_0\}^T \tag{5.4-6}$$

where p_o is the initial value of p obtained from Eqs. (5.4-2), (5.4-3) and (5.4-5). Applying the algebraic transformation

$$\bar{v} = v - v_0\,, \tag{5.4-7}$$

equation (5.4-4) can be expressed as

$$\bar{v}' = H\bar{v} + \hat{q} \tag{5.4-8}$$

with the initial conditions

$$\bar{v}\big|_{t=t_1} = 0\,. \tag{5.4-9}$$

where $\hat{q}$ =Hv_0+q.

Applying the DQ matrix formulation presented in section 5.2 (e.g. equation (5.2-2)) to equation (5.4-8), we have

$$V\bar{A}^T - HV = Q \tag{5.4-10}$$

where $\bar{A}$ is obtained by eliminating the first row and column of the DQ weighting coefficient matrix for the 1st order derivative. Capital letter V and Q denote rectangular matrix stacked from vector $\bar{v}$ and $\hat{q}$ as in equations (5.2-3) and (5.2-4). Obviously, Eq. (5.4-10) is a Lyapunov-like algebraic matrix equation.

5.4.2. Formulation for the Problems with Proportional Damping or No Damping

If only proportional damping is included in equation (5.4-1) or damping is neglected. the problems will be simplified greatly. Rayleigh damping using proportional damping has been in use and has the form

$$C = \alpha M + \beta K, \tag{5.4-11}$$

where α and β are the given Rayleigh constants. Thus, the equation (5.4-11) becomes

$$M(u'' + \alpha u') + K(u' + \beta u) = f \tag{5.4-12}$$

In order to use the matrix formulation presented in section 5.2, we must convert the initial conditions (Eqs. (5.4-2) and (5.4-3)) of all elements in unknown vector u into certain same constant. An algebraic transformation is presented by

$$\bar{u} = u - d - ht + ht_1. \tag{5.4-13}$$

Thus,

$$\bar{u}' = u' - h \tag{5.4-14}$$

$$\bar{u}'' = u'' \tag{5.4-15}$$

Substituting Eqs. (5.4-13), (5.4-14) and (5.4-15) into Eqs. (5.4-1), (5.4-2), and (5.4-3), we have

$$M(\bar{u}'' + \alpha \bar{u}') + K(\beta \bar{u}' + \bar{u}) = \bar{f} \tag{5.4-16}$$

$$\bar{u}\Big|_{t=t_1} = 0 \tag{5.4-17}$$

$$\bar{u}'\Big|_{t=t_1} = 0 \tag{5.4-18}$$

where $\bar{f} = f - Ch - Kd - Kht + Kht_1$.

Before the formulation of equation (5.4-16), we first consider the approach applying the initial conditions in the DQ method for the problems involving the derivative of no less than the 2nd order, which is very important for computational efficiency. We extend the DQU and DQZ approaches for boundary value problems in section 4.4 to handle the present initial value problems. Considering the initial value equation (5.4-17), the DQ formulation for the 1st order derivative can be expressed in matrix as

$$\begin{bmatrix} 0 & A_{12} & \cdots & A_{1n} \\ 0 & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & A_{n2} & \cdots & A_{nn} \end{bmatrix} \begin{Bmatrix} \bar{u}_1 \\ \bar{u}_2 \\ \vdots \\ \bar{u}_n \end{Bmatrix} = \begin{Bmatrix} \bar{u}_1' \\ \bar{u}_2' \\ \vdots \\ \bar{u}_n' \end{Bmatrix}, \tag{5.4-19}$$

namely,

$$\bar{A}\{\bar{u}\} = \{\bar{u}'\} \tag{5.4-20}$$

where $\bar{A}$ is the modified weighting coefficient matrix by using the initial value equation (5.4-17). Similarly, we have

$$\bar{A}\{\bar{u}'\} = \{\bar{u}''\} \tag{5.4-21}$$

by using the initial value equation (5.4-18). Thus,

$$\{\bar{u}''\} = \bar{A}\{\bar{u}'\} = \bar{A}\bar{A}\{\bar{u}\} = \bar{B}\{\bar{u}\} \tag{5.4-22}$$

$\overline{A}$ and $\overline{B}$ are the resulting modified weighting coefficient matrices for the 1st and 2nd order derivatives, respectively, in which the initial conditions have been taken into account. In terms of the modified weighting coefficient matrices $\overline{A}$ and $\overline{B}$, the DQ formulation for equation (5.4-16) can be expressed as

$$M\overline{U}\left(\overline{B}^{T} + \alpha\overline{A}^{T}\right) + K\overline{U}\left(\beta\overline{A}^{T} + I\right) = \overline{F}, \tag{5.4-23}$$

where $\overline{U}$ and $\overline{F}$ are n×(m-1) rectangular matrices if m grid points are used in the DQ method. The above equation can be converted into a Lyapunov-like algebraic matrix equation, namely,

$$\overline{U}G + R\overline{U} = Q \tag{5.4-24}$$

where $G = \left(\overline{B} + \alpha\overline{A}\right)^{T}\left(\beta\overline{A}^{T} + I\right)^{-1}, R = M^{-1}K, \qquad Q = M^{-1}\overline{F}\left(\beta\overline{A}^{T} + I\right)^{-1}$.

Similar to the DQZ approach for handling boundary conditions discussed in section 4.4, the initial equation (5.4-18) can be approximated by

$$\sum_{j=1}^{m} A_{1j}\bar{u}_{j} = 0. \tag{5.4-25}$$

where u$_j$'s are corresponding to the displacement at the jth time point. We have

$$\bar{u}_{2} = -\frac{1}{A_{12}}\sum_{j=3}^{m} A_{1j}\bar{u}_{j}. \tag{5.4-26}$$

Substituting the initial equation (5.4-17) and equation (5.4-26) into the DQ formulations for the first and second derivatives, respectively, we have

$$\vec{u} = \hat{A}\vec{u} \tag{5.4-27}$$

$$\vec{u}'' = \hat{B}\vec{u} \tag{5.4-28}$$

where $\vec{u} = \left\{u_{3} \quad u_{4}, \quad \ldots, \quad u_{m}\right\}$. Note that $\hat{A}$ and $\hat{B}$ are (m-2)×(m-2) modified weighting coefficient matrices different from (m-1)×(m-1) dimension $\overline{A}$ and $\overline{B}$ in equation (5.4-24). It is emphasized that the DQ formulations at the first two grid points are omitted here. Using the modified $\hat{A}$ and $\hat{B}$, the resulting formulation for equation (5.4-16) are

$$\overline{U}G + R\overline{U} = Q, \tag{5.4-29}$$

where $G = \left(\hat{B} + \alpha\hat{A}\right)^{T}\left(\beta\hat{A}^{T} + I\right)^{-1}$, $Q = M^{-1}\overline{F}\left(\beta\hat{A}^{T} + I\right)^{-1}$ and $R = M^{-1}K$. Matrices $\overline{U}$ is generated form vector $\vec{u}$ as in equations (5.2-2). Eq. (5.4-29) is also an algebraic Lyapunov matrix equation.

Eqs. (5.4-10), (5.4-24) and (5.4-29) are Lyapunov-like algebraic matrix equation, but it should be pointed out that the size of equation (5.4-10) is 2n×(m-1), while the size of the equation (5.4-24) and (5.4-29) are n×(m-1) and n×(m-2), respectively. The advantages of equation (5.4-10) is to evaluate simultaneously the velocity and displacement vector. However, for the problems with proportional damping or no damping, equation (5.4-24) or (5.4-29) is still preferred. Also, the DQZ approach requires less computing effort than the Wang and Bert's DQU approach, but as will be seen later, the former computational accuracy is lower than the latter. It is noted that no additional equations are required in the solution of equations (5.4-10), (5.4-24) or (5.4-29) since the initial conditions have taken into account. Finally, the obtained displacement values can be used to calculate the corresponding velocity and acceleration by using equations

$$\overline{U}' = U\overline{A}^{T}, and \quad \overline{U}'' = U\overline{B}^{T} \tag{5.4-30}$$

in the DQU approach and

$$\overline{U}' = U\hat{A}^T, and \quad \overline{U}'' = U\hat{B}^T \tag{5.4-31}$$

in the DQZ approach.

5.4.3. Applications

Example 1. We consider a simple system of two differential equations provided by Bathe and Wilson (1976) as a test example. They are given by

$$M\ddot{x} + Kx = q \tag{5.4-32}$$

with the initial conditions

$$x_1(0) = x_2(0) = 0, \qquad \dot{x}_1(0) = \dot{x}_2(0) = 0\,. \tag{5.4-33}$$

in which $M = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \qquad K = \begin{bmatrix} 6 & -2 \\ -2 & 4 \end{bmatrix}, \qquad q = \begin{Bmatrix} 0 \\ 10 \end{Bmatrix}, \qquad x = \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix}.$

Applying the linear transformation equation (5.4-13), we have

$$M\ddot{y} + Ky = p\,, \tag{5.4-34}$$

where $y = x - d - ht + ht_1$, d, h and t_1 are the initial displacement vector, velocity vector and time at each time step. $p = q - Kd - Kht + Kht_1$. The DQ formulation using the DQU approach for this case can be expressed as

$$MY\overline{B} + KY = P\,. \tag{5.4-35}$$

Furthermore,

$$Y\overline{B} + M^{-1}KY = M^{-1}P\,, \tag{5.4-36}$$

where $\overline{B}$ is the modified DQ weighting coefficient matrix for the second order derivative as defined in equation (5.4-22). Y and P are rectangular matrix. Eq. (5.4-36) is a Lyapunov-like matrix equations. The resulting DQ formulation using the DQZ approach for this case is $Y\hat{B}^T + M^{-1}KY = M^{-1}P$. (5.4-37)

where $\hat{B}$ is defined in equations (5.4-28). However, It is noted that in the following the DQ results are obtained using equation (5.4-36) unless otherwise indicated.

Let grid interval Δt=0.28, we compute this example using eleven equally spaced grid points. Table 5-1 lists the results by the DQ, Wilson θ (θ=1.4) and Newark (α=0.25, δ=0.5) methods under the same Δt=0.28. Note that the grid interval Δt in the DQ method is equivalent to the time step in the Wilson θ and Newmark methods but different from the step size in the present DQ method. For example, the present step size in the DQ method is (11-1)×Δt=2.8.

Figs. 5-1 and 5-2 display the relative errors for the displacement x_1 and x_2 using the DQ method as well as the Wilson θ (θ=1.4) and Newmark (α=0.25, δ=0.5), and Houblot methods under Δt=0.28. The DQ method employs eleven equally spaced grid points and the Chebyshev grid points under the comparable step size 2.8. The relative error is defined as the ratio of the absolute error to the absolute value of the analytical answer. It is observed that the Wilson θ, Newmark and Houblot methods have basically the same computational efficiency. In contrast, the DQ method have much high accuracy. Except for the solutions at the first four grid points, the DQ method yields about 100 times smaller errors than these conventional integration methods. This is due to the fact that the DQ method has high order convergence rate (nine order accuracy using eleven grid points), while other methods are only two order accuracy. The results indicated by DQ* in Fig. 5-1 and 5-2 are computed by the DQ method using the DQZ approach. It is obvious that the DQ results using Wang and Bert's DQU approach have higher accuracy. In addition, as is expected, the DQ method using the Chebyshev grids also yields more accurate results than using the equally spaced grids. We also compute 1000 steps for the case under Δt=0.28, the DQ method still maintains stable. If choosing the large step size, say Δt=28, no unstable behavior occurs in the DQ method.

Table 5-2 lists the relative errors of the DQ solutions using eleven and twenty-one the Chebyshev grid points under the step size 2.8, respectively. err1 and err2 denote the relative errors corresponding to displacement x_1 and x_2, respectively. It is noted that the results using 21 grid points are more accurate than using 11 grid points. Therefore, more grid points, more accurate results are obtained. It is concluded that the larger number of grid points should be used for the system of high order. However, for the stiff problems, more care should be paid to choose the number of grid points and grid intervals because the stiff systems are characterized by irregularly shaped curves particularly in stiff regions (Burka, 1982).

Example 2. The tow-degree-of-freedom system with nonproportional damping (D’souza and Garg, 1984) is given by

$$M\ddot{x} + C\dot{x} + Kx = q \tag{5.4-38}$$

where x={x_1, x_2}^{T}.

$$M = \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}, \qquad C = \begin{bmatrix} 0.2 & -0.1 \\ -0.1 & 0.1 \end{bmatrix}, \qquad K = \begin{bmatrix} 21 & -1 \\ -1 & 1 \end{bmatrix}, \qquad q = \begin{Bmatrix} 0 \\ 4 \end{Bmatrix}$$

The initial conditions are

$$x(0) = \dot{x}(0) = 0. \tag{5.4-39}$$

By using the linear transformations (5.4-5) and (5.4-7), we have

$$\bar{v}' = H\bar{v} + \hat{q}, \tag{5.4-40}$$

where

$$H = \begin{bmatrix} -0.1 & 0.05 & 1 & 0 \\ 0.005 & -0.005 & 0 & 0.1 \\ -20.9875 & 0.9925 & -0.1 & 0.005 \\ 0.99475 & -0.99725 & 0.05 & -0.005 \end{bmatrix}$$

The details on $\bar{v}$ and $\hat{q}$ see section 5.4.1. The matrix formulation for this case is

$$V\bar{A}^T - HV = Q\ , \tag{5.4-41}$$

where $\bar{A}$ is the modified DQ weighting coefficient matrix for the 1st order derivative as defined in equation (5.4-17). Q is a rectangular matrices stacked from $\hat{q}$. Equation (5.4-41) is also a Lyapunov matrix equation.

D’souza and Garg (1984) found that, when the time step is increased to 0.5s, the Newmark beta (α=0.5, β=1/6), central difference predictor, two-cycle iteration with trapezoidal rule and the fourth-order Runge-Kutta schemes yielded unstable solutions for this case, while Park (Prak, 1975), Houbolt, and Wilson θ (θ=1.5) schemes remain stable. The present DQ method is also stable under Δt=0.5s or more when eleven or twenty-one equally spaced and the Chebyshev grid points are employed, and the results using eleven equally spaced grid points under larger Δt=2.5s (step size 25s) are shown in Fig. 5-4 and 5-6 for the displacement x_1 and x_2, respectively. The relative errors at t=250s for displacement x_1 and x_2 are 1%. In D’souza and Garg (1984) the Wilson, Prak and Houbolt methods are applied for this case under Δt=0.05, 0.1, respectively. However, the numerical results are not presented there and thus a direct accuracy comparison can not be given. The analytical solutions of displacement for this case are plotted in Fig. 5-3 for x_1 and Fig 5-5 for x_2. More example study may prove beneficial. In the above two examples at least, the DQ method proves very successful.

In the present study, we consider only tow-degree-of-freedom system. Thus, the computational effort seems to be a little larger in the DQ method using 11 eleven grid points than in the Newmark, Wilson θ and Houbolt methods under comparable step size. The ratio of computing effort is 1+(m-1)/n, n and m denote the order of systems and grid points in the DQ method. However, as discussed in section 5.4.2, the computing effort in the DQ method is nearly the same as those in the Newmark and Wilson methods under comparable step size for the larger systems in engineering practice, say that eleven grid points are used in the DQ method for the analysis of the system more than 50 degree of freedom. In addition, by using less grid points or larger time step, the DQ method consumes a comparable computing effort for these cases in comparison to other methods while still producing very accurate solutions. Therefore, it is believed that the DQ method has much higher computational efficiency than all other methods involved here.

Table 5–I. Numerical results for example I under Δt=2.8

Method	Analytical		DQ		Wilson θ		Newmark	
	x_1	x_2	x_1	x_2	x_1	x_2	x_1	x_2
2Δt	0.0381	1.412	0.0389	1.410	0.0525	1.34	0.0504	1.35
4Δt	0.486	4.094	0.486	4.093	0.490	3.92	0.485	4.00
6Δt	1.657	5.291	1.656	5.292	1.54	5.31	1.58	5.34
8Δt	2.861	4.277	2.860	4.278	2.67	4.61	2.76	4.48
10Δt	2.806	2.806	2.806	2.806	2.82	3.06	2.85	2.90
12Δt	1.157	2.488	1.159	2.485	1.54	2.29	1.40	2.31

Table 5–2. The relative errors of the DQ results for example I using the Chebyshev grid spacings (step size=2.8)

Sequence of grid point	11 grid points		21 grid points	
in time direction	err1	err2	err1	err2
2	1.4E-2	2.5E-4	6.5E-3	1.8E-5
4	1.3E-4	2.6E-5	6.0E-5	1.8E-6
6	4.1E-5	2.8E-5	1.1E-6	1.8E-7
8	1.3E-5	2.0E-5	1.2E--6	5.4E-7
10	6.4E-6	1.6E-5	1.2E-6	1.4E-6
12	7.0E-5	1.1E-4	2.4E-6	2.7E-7

5.5. Stiff Problems

The stiff ordinary differential equations are often encountered in automatic control, electronic network, biosciences, physics and chemical kinetics. The key to handle stiff problems is the numerical stability of methods. However, most explicit integration formulas have a bounded region of stability. For example, the well-known fourth-order explicit Runge-Kutta method has a region of stability, and very small step sizes are required by this method. These methods are, in general, inefficient for solving stiff systems as they do not properly simulate the rapidly decaying solutions. The stability obstacle can be overcome by considering implicit Runge-Kutta methods. However, such methods suffer from a serious practical disadvantage in that the computing effort is in general costly for large systems. Some effective solution techniques have been also well developed to handle stiff systems (Gear, 1967, 1971a, 1971b). Based on the foregoing analysis of the DQ method, the purpose of this section is to provide several numerical examples for demonstrating the efficiency in the DQ solution of initial value problems for stiff systems of ordinary differential equations.

Example 1. $y' = -\alpha y + \beta, \quad y(0) = 0$ (5.5-1)

The accurate solution for this case is

$$y = \frac{\beta}{\alpha} - \frac{\beta}{\alpha} e^{-\alpha t}. \tag{5.5-2}$$

The linear transformation

$$u = y - y(t_1) \tag{5.5-3}$$

is applied to equation (5.5-1). t_1 means the initial time at each iteration step. We have

$$u' = -\alpha u - \alpha y(t_1) + \beta, \quad u(t_1) = 0. \tag{5.5-4}$$

In terms of the DQ method, we have

$$\bar{A}\vec{u} = -\alpha\vec{u} - \alpha y(t_1) + \beta, \tag{5.5-5}$$

where $\bar{A}$ is the modified weighting coefficient matrix for the 1st order derivative. The equation can be restated as

$$[\bar{A} + \alpha I]\vec{u} = -\alpha y(t_1) + \beta \tag{5.5-6}$$

The Gauss elimination method is used to solve the above algebraic equations. Let parameters α=1000, β=10^5, and time step h=0.1, the relative error at t=1, 2, 3, 4, 5, and 6 are listed in table 5-3. It is found that the relative errors after the sixth step are zero. Let α=10^5, β=10^7, the relative errors after h=4 become zero. In the above two cases, stiffness are 1000 and 10^5, respectively. Thus, the transients dies out very quickly. It is concluded that the DQ method is very safe and reliable in the solution of stiff systems.

Table 5-3. The relative errors at the first six steps (α=1000, β=10^5, and h=0.1)

time t	1	2	3	4	5	6
relative error	9.2E-4	8.4E-7	7.7E-10	7.9E-13	7.1E-16	0.0

Example 2. This example was given in many standard textbooks. Therefore, it is a benchmark problem for testing numerical algorithms for stiff ordinary differential equations.

$$\begin{cases} \dfrac{du}{dt} = 998u + 1998v \\ \dfrac{dv}{dt} = -999u - 1999v \end{cases}. \tag{5.5-7}$$

The corresponding initial conditions are

$$u(0) = 1, \quad v(0) = 0. \tag{5.5-8}$$

Apply the linear transformation

$$\bar{u} = u - u(t_1) \quad \text{and} \quad \bar{v} = v - v(t_1) \tag{5.5-9}$$

to equation (5.5-7), we can obtain the following DQ formulation:

$$X\bar{A}^T = RX + C \tag{5.5-10}$$

where

$$X = \begin{bmatrix} \overline{u}_2 & \overline{u}_3 & \cdots & \overline{u}_N \\ \overline{v}_2 & \overline{v}_3 & \cdots & \overline{v}_N \end{bmatrix},\ R = \begin{bmatrix} 998 & 1998 \\ -999 & -1999 \end{bmatrix},\ C = \begin{bmatrix} c_1 & c_1 & \cdots & c_1 \\ c_2 & c_2 & \cdots & c_2 \end{bmatrix},$$

$$c_1 = 998u(t_1) + 1998v(t_1), \quad c_2 = -999u(t_1) - 1999v(t_1).$$

The accurate solutions for this case are

$$u(t) = 2e^{-t} - e^{-1000t} \tag{5.5-11}$$

and

$$u(t) = 2e^{-t} - e^{-1000t} \tag{5.5-12}$$

The stiffness in the example is 1000. Similar to the situations in section 5.4 for structural dynamic problems, the BS approach is exploited for solving the formulation equation (5.5-10). The DQ method is stiff stable even if the larger time step h=0.5 is used. In contrast, the time step h in applying the Runge-Kutta method for this case can not be larger than 0.00278. Therefore, the Runge-Kutta method is obviously not applicable for the stiff problems. The relative error of the DQ solutions using eleven equally spaced grid points under h=0.1 is no more than 10^{-9}. Therefore, the DQ method is shown to have rather high accuracy. We also calculate other some stiff systems, and the same conclusions can be drawn.

5.6. Nonlinear Problems

The nonlinear initial value problems are often encountered in practice. In this section we investigate the potential of the DQ method in the solution of nonlinear initial value problems. The Hadamard product and SJT product technique are used for formulation and calculation of the respective Jacobian matrix. Unfortunately, however, the fast algorithms for the Lyapunov matrix equation, which are very efficient for linear initial value problems, are not applicable for solving the Newton iteration equation resulting from applying the Newton-Raphson method to the DQ nonlinear formulations. Thus, the computing effort is obviously increased. In the following section 5.7 we will discuss some possible routes to improve the computational efficiency.

A combined use of the DQ and Adams-Moulton methods (Bellman et al, 1972) were exploited to analyze some nonlinear initial-boundary problems. The partial derivative in spatial dimensions were approximated by using DQ method, while the derivative in time was analogized by an Adams-Moulton method. We recalculate two examples of them only by using the DQ method, e.g., the following examples 1and 2. Example 3 is a nonlinear structural dynamic systems.

Example 1. $u_t(x,t) = uu_{,x}(x,t)$, 0∠x≤1, 0≤t≤T; $u(x,o) = 0.1x$ (5.6-1)

The above equation is a hyperbolic nonlinear one. First, let

$$v(x,t) = u(x,t) - u(x,t_1) \tag{5.6-2}$$

When t_1=0, u(x, t_1)=0.1x. Therefore,

$$v_t(x,t) = [v(x,t) + u(x,t_1)][v_x(x,t) + u(x,t_1)] \tag{5.6-3}$$

and

$$v(x,t_1) = 0. \tag{5.6-4}$$

In terms of the DQ method, equation (5.6-3) can be approximated by

$$V\overline{A}_t^T = (V + U(x,t_1)) \circ (A_x V + A_x U(x,t_1)). \tag{5.6-5}$$

where A_x is the M×M DQ weighting coefficient matrix for spatial derivative. M is the number of grid points used in x direction, while N is the number of grid points along time direction. $\overline{A}_t$ is the (N-1)×(N-1) modified DQ weighting coefficient matrix for time derivative.

$$U(x,\ t_1)\ =\begin{bmatrix} u(x_1,\ t_1) & u(x_1,\ t_1) & \cdots & u(x_1,\ t_1) \\ u(x_2,\ t_1) & u(x_2,\ t_1) & \cdots & u(x_2,\ t_1) \\ \vdots & \vdots & \cdots & \vdots \\ u(x_M,\ t_1) & u(x_M,\ t_1) & \cdots & u(x_M,\ t_1) \end{bmatrix}_{M\times(N-1)} \tag{5.6-6}$$

Note that the Hadamard product is applied to formulate equations (5.6-3) into DQ analogue equation (5.6-5). By applying the Kronecker product to equation (5.6-5), we have

$$\psi\{\vec{V}\}\ =\ \left(I_M\ \otimes\ \overline{A}_t\right)\vec{V}\ -\ \left(\vec{V}\ +\ \vec{U}\right)\ \circ\ \left(\left(A_x\ \otimes\ I_{(N-1)}\right)\vec{V}\ +\ \left(A_x\ \otimes\ I_{(N-1)}\right)\vec{U}\right) \tag{5.6-7}$$

Employing the SJT product, we can obtain the Jacobian matrix of the above equation:

$$\frac{\partial\psi\{\vec{V}\}}{\partial\vec{V}}\ =\ I_M\ \otimes\ \overline{A}_t\ -\ I_{M(N-1)}\Diamond\left[\left(A_x\ \otimes\ I_{(N-1)}\right)\vec{V}\ +\ \left(A_x\ \otimes\ I_{(N-1)}\right)\vec{U}\right]\ -\ \left(A_x\ \otimes\ I_{(N-1)}\right)\Diamond\left(\vec{V}\ +\ \vec{U}\right) \tag{5.6-8}$$

The exact solution for the case is given by

$$u(x,\ t)\ =\ x\big/(t\ -\ 10)\,. \tag{5.6-9}$$

Eleven equally spaced grid points and seven Legender points are used, respectively, along x-and y-directions. The time step is h=0.1. The relative errors of the DQ solutions are listed in Table 5-4. It is observed that the DQ method yields very high accurate solutions.

Table 5-4. The relative errors of differential quadrature solutions for example 1.

t	0.1	0.1	0.1	0.5	0.5	0.5	1.0	1.0	1.0
x	x_1	x_4	x_7	x_1	x_4	x_7	x_1	x_4	x_7
error	1.5E-8	1.5E-8	1.5E-8	1.6E-8	1.6E-8	1.6E-8	1.7E-8	1.7E-8	1.7E-8

Example 2. The Burger's equation is stated as

$$u_{,t}\ +\ uu_{,x}\ =\ \nu u_{,xx}\,,\qquad \nu>0, \tag{5.6-10}$$

where the term $uu_{,x}$ represents a nonlinear convective term while $\nu u_{,xx}$ means a dissipative term. The parameter ν represents the inverse of the Reynolds number, which determines the importance of convection versus that dissipation. Equation (5.6-10) serves as a useful model since it possesses features in common with the Navier-Stokes equation and has become a popular benchmark for testing numerical technique to partial differential equations.

Initial value condition for equation (5.6-10) is

$$u(0,\ x)\ =\ f\,(x)\,. \tag{5.6-11}$$

In this case we choose

$$f\,(x)\ =\ -2\upsilon\left[b\pi\ cos\ \pi x\ +\ c\pi\ cos\ \pi x/2\right]\big/\left[b\ sin\ \pi x\ +\ c\ sin\ \pi x/2\ +\ d\right]. \tag{5.6-12}$$

where constants ν=0.01, b=0.2, c=0.1, d=0.3.

The analytic solution for this case is

$$u(t.\ x) = -2\upsilon\left[b\pi e^{-\upsilon\pi^2 t} \cos \pi x + \frac{\pi c}{2} e^{-\upsilon\pi^2 t/4} \cos \pi x/2 \right] \Big/ \left[b e^{-\upsilon\pi^2 t} \sin \pi x + c e^{-\upsilon\pi^2 t/4} \sin \pi x/2 + d \right] \tag{5.6-13}$$

The same grid spacings and time step as example 1 are used for this case. Table 5-5 displays the relative errors of the DQ solutions. The DQ method is demonstrated again to be a high accurate approach. We also compute the cases in some different situations by choosing large the Reynolds number, namely 10^3, 10^5, and 10^7, respectively. The accuracies of the DQ method are even higher.

Table 5-5. The relative errors of differential quadrature solutions for example 2.

t	0.1	0.1	0.1	0.5	0.5	0.5	1.0	1.0	1.0
x	x_1	x_4	x_7	x_1	x_4	x_7	x_1	x_4	x_7
error	1.3E-4	1.1E-4	1.2E-4	1.6E-3	3.7E-4	6.4E-4	3.8E-3	4.2E-4	8.8E-4

Example 3. This example was considered by D'Souza and Garg (1984). The dynamic equations are given by

$$\begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix} \begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{Bmatrix} + \begin{bmatrix} 0.2 & -0.1 \\ -0.1 & 0.1 \end{bmatrix} \begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{Bmatrix} + \begin{bmatrix} 21 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + 0.5\left(x_2 - x_1\right)^2 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 4 \end{Bmatrix} \tag{5.6-14}$$

The initial conditions are expressed as

$$x(0) = \dot{x}(0) = 0 \tag{5.6-15}$$

The dynamic equations can be restated in the Hadamard product form as

$$M\ddot{x} + C\dot{x} + Kx + \left(K_2 x\right)^{\circ 2} \circ \left(K_3 x\right) = q \tag{5.6-16}$$

where

$$K_2 = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}, \qquad K_3 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

The detailed solution procedure is omitted for the brevity. The DQ solutions are displayed in Figure 5-7, and agree well with those by using other numerical techniques (D'Souza and Garg 1984).

5.7. Discussions

The DQ method has been extensively used to approximate the spatial derivative so far. However, like the pseudo-spectral and collocation methods, the method seems to be not very suitable for problems with complex geometries. Although the coordinate mappings and multidomain approaches can be used to overcome this drawback, there is a larger loss in the efficiency and simplicity of the DQ method. As is shown in the foregoing study, the shortcoming are not inherent in applying the DQ method to the structural dynamic problems and stiff initial value problems. The principal contributions of the work are to present the matrix formulation in the DQ method for ordinary differential equations of initial value systems. Therefore, the resulting formulation are reduced to an algebraic Lyapunov matrix equation, and the formulation and computational effort are greatly abbreviated to a comparable level with other conventional methods while the high accuracy and ease of implementation in the DQ method yet maintains. We also point out that the DQ method is an A-stable method. To our knowledge, this fact has never been indicated in the literature. The DQU approach proposed by Wang and Bert (1993a) is extended to the initial value problems and shows high efficiency. The DQ method using the presented techniques can be also efficiently used to solve other initial-boundary problems, for example, the transport problems (1984b). Based on the given numerical comparisons and theoretical analysis, it is concluded that the DQ method is a promising numerical technique to linear dynamic analysis and stiff initial value problem.

The following points should be also pointed out:

1) Many studies (Bellman et al. 1972, 1974a, 1979; Mingle, 1973, 1977; Quan et al., 1989; Shu et al., 1992, Malik et al, 1994; Bert et al., 1989; Striz et al., 1988) have shown that the DQ method is especially efficient for the nonlinear boundary value problems. Therefore, it is very significant to apply the DQ method to nonlinear structural dynamic problems.
2) It is also well known that the A-stable methods are strong stability and reliable for the stiff problems (Gear, 1971). Since the DQ method is A-stable. it is also expected that the DQ method will be an efficient technique to handle a variety of the stiff equations.
3) The adaptive mesh technique can be easily used in the DQ method for the present purpose.
4) The accuracy and computing efficiency of the DQ method also depend greatly on the proper choice of the trial functions as in the collocation method (Bert et al., 1993; Wang and Wang, 1994; Striz et al., 1995). For example, it is expected that the DQ method using the trigonometric functions may be more efficient than using the present polynomial basis functions for the periodic dynamic problems, especially for the problems with high order frequency, while split range functions (Mansell, 1993) or other proper test function (Chang, 1993) may obviously improve the efficiency in applying the DQ method for the initial value problems involving steep gradients.

In what follows we discuss the application of the DQ method to nonlinear stiff initial value problems. As was pointed out in section 5.6, as soon as the equation is allowed to have variable coefficient or nonlinear operators, the computational effort will be obviously increased.

Considering example I in section 5.6, the Newton iteration equations for this case can be stated as

$$\vec{V}^{(k+1)} = \vec{V}^{(k)} - \left[\frac{\partial\psi\left(\vec{V}^{(k)}\right)}{\partial\vec{V}^{(k)}}\right]^{-1}\psi\left(\vec{V}^{(k)}\right). \tag{5.7-1}$$

The above equations can be converted into a set of algebraic equations, namely,

$$\frac{\partial\psi\left\{\vec{V}^{(k)}\right\}}{\partial\vec{V}}\left[\vec{V}^{(k+1)} - \vec{V}^{(k)}\right] = \psi\left(\vec{V}^{(k)}\right). \tag{5.7-2}$$

Let $\delta\vec{V}^{(k+1)} = \vec{V}^{(k+1)} - \vec{V}^{(k)}$, we have

$$\frac{\partial\psi\left\{\vec{V}^{(k)}\right\}}{\partial\vec{V}}\delta\vec{V}^{(k+1)} = \psi\left\{\vec{V}^{(k)}\right\}. \tag{5.7-3}$$

Substituting equation (5.6-8) for computing Jacobian matrix into the above equation, we have

$$\begin{aligned}\delta V^{(k+1)}\bar{A}_t - \left(\bar{A}_x V^{(k)} + \bar{A}_x U^{(k)}\right)\circ\delta V^{(k+1)} - \\ \left(V^{(k)} + U^{(k)}\right)\circ\left(\bar{A}_x\delta V^{(k)}\right) = \psi\left\{V^{(k)}\right\},\end{aligned} \tag{5.7-4}$$

where $\delta V^{(k)}$, $V^{(k)}$, and $U^{(k)}$ are rectangular matrices. The Newton iteration equation (5.7-4) is a linear algebraic equations with the Hadamard product operation. In the derivation of equation (5.7-4), we employ the following property of the SJT product:

$$\vec{C} \circ \left(A\vec{V} \right) = \left(A \Diamond \vec{C} \right)\vec{V} , \tag{5.7-5}$$

where $\vec{C}$ is the constant vector. This property of the SJT product is also applied in section 2.9.1 for deriving the formulation-S for systems with varying coefficients.

Eq. (5.7-4) are not different from the traditional Lyapunov matrix equation due to its Hadamard product operation. The fast algorithms for the latter are not applicable for solving matrix equation of such type. Therefore, Eq. (5.7-4) must be converted into the standard algebraic equation as done in section 5.6 if the Gauss elimination method is used. Although the very high accuracy of the DQ method can compensate to some extent for great computational effort, there is still much room for developing some efficient techniques for abbreviating computational effort. Three approaches for this purpose are presented as follows.

In order to ensure the convergence, the very small control parameter in the standard iteration formula of the simple iteration method is usually required for the stiff initial value problems and, thus, the computing efficiency is in general very poor. It is well known that the Newton-Raphson method must be used to solve such problems. In section 2.2, we found that the DQ formulations in Hadamard product form can easily yield some very efficient iteration formulas of the simple iteration method by using the Hadamard matrix function, power and inverse concepts. Moreover, the most effective iteration formula can be easily determined in advance by using the SJT product. Some boundary value problems such as the example discussed in section 2.2, which are similar to stiff initial value problems and need very high computing cost to use the standard iteration formula, are efficiently solved by using new iteration formula obtained via the Hadamard matrix inverse approach. It is hoped that these techniques can be extended to handle stiff initial value problems. It is also expected that the computational effort in the DQ method using this technique for nonlinear stiff problems is nearly the same as other existing methods such as Gear method while the high accuracy of the DQ method is still maintained. Next, section 2.5 discussed that the Hadamard product and SJT product approach can be applied for decoupling computation of nonlinear boundary value problems. The extension of the decoupling computations is another hopeful route to reduce computing effort for nonlinear stiff problems of large scale. Finally, we hope to develop a Gauss-Sidel iteration method for the efficient solution of the Newton iteration matrix equation with the Hadamard product operation. The computing effort using the technique will be greatly reduced in comparison to the Gauss elimination method for the same task. A combined use of the above three approaches may greatly reduce the computing effort in applying the DQ method for nonlinear stiff initial value problems and makes the DQ method very competitive to other numerical techniques.

CHAPTER 6.
STRUCTURES OF WEIGHTING COEFFICIENT MATRICES

6.1. Introduction

The DQ and HDQ weighting coefficient matrices play a prominent role in the application of the DQ and HDQ methods. However, the research in this field is neglected in the literature. Quan and Chang (1989a) pointed out that the skew centrosymmetric and centrosymmetric structures of the DQ weighting coefficient matrices for the 1st and 2nd order derivatives, and this properties can be used to decompose the DQ coefficient matrix for ordinary differential equations of systemic models into two smaller size matrices. But in that reference they did not use the centrosymmetric and skew centrosymmetric terminologies and, thus, can not be fully aware of the possible advantages in application of these properties. Chen and Yu (1993) pointed clearly out the relation between the DQ weighting coefficient matrices and centrosymmetric or skew centrosymmetric matrices.

In this chapter, we prove that the structures of the weighting coefficient matrices for any order derivative in the DQ and HDQ methods are either centrosymmetric or skew centrosymmetric if the grid spacing is symmetric with respect to the center point irrespective whether the grid spacings are equal or nonequal. It is known that, in the evaluation of the determinant, inverse and eigenvalue and eigenvectors, a centrosymmetric matrix can be factorized into two smaller size matrices. By applying these properties, the multiplication complexity can be reduced by 75% and the efficiency of the DQ and HDQ methods can be significantly increased, which is also demonstrated by solving the free vibration of beams and plates in this Chapter.

Secondly, the skew centrosymmetric matrix is defined and its properties are found to be similar to those of the centrosymmetric matrices. It is shown that this kind of matrix is related to the DQ and HDQ weighting coefficient matrices for all odd order derivatives. Third, the reducibility using the centrosymmetric properties in the DQ and HDQ methods is extended to non-systemic problems. Next, the properties of the Hadamard product and SJT product involving the centrosymmetric matrix are discussed. The reducibility for the nonlinear problems with symmetric structures is obtained by using these properties in the DQ-type methods. Finally, some conclusions are drawn based on the results reported herein.

6.2. Centrosymmetric Structures

This section shows that the weighting coefficient matrices of the DQ and HDQ methods are either centrosymmetric or skew centrosymmetric depending on the order of the corresponding derivatives. Some properties of these matrices are briefly stated or discussed.

6.2.1. Centrosymmetric structure of weighting coefficient matrices in the DQ method

First, the definitions about centrosymmetric and skew centrosymmetric matrices are given

Definition 6.2.1: A N×N matrix $Q=[q_{ij}]$ is centrosymmetric if

$q_{ij=}\ q_{N+1-i,N+1-j}$, I, j=1, 2, ..., N. (6.2-1)

Then, Q can be characterized by

$JQJ=Q$ (6.2-2)

where J denotes the contra-identity matrix given by

$$J = \begin{bmatrix} 0 & \dots & 0 & 1 \\ 0 & \cdots & 1 & 0 \\ \vdots & .\cdot^{\cdot} & \vdots & \vdots \\ 1 & \cdots & 0 & 0 \end{bmatrix} \qquad (6.2\text{-}3)$$

which has unit elements along the secondary diagonal and zeros elsewhere, noting that J^T=J and J^2=I, the unit matrix, The effect of premultiplying any matrix by J reverses the order of its rows, and postmultipling reverses the order of its columns.

Definition 6.2.2: A new, skew centrosymmetric matrix R=[$r_{ij}]_{N\times N}$ with the following property, is introduced:

$r_{ij} = -r_{N+1-i,N+1-j}$, i,j=1,2,....,N. (6.2-4)

and

R=–JRJ (6.2-5)

To our knowledge, this is a new form of matrix, which has not been reported previously.

The grid spacings often used in the DQ method are symmetric such as equally spaced grids and the zeros of the shifted Legendre or the shifted Chebyshev polynomials. For symmetric grid spacings, over a domain 0≤ x ≤1, it is true that:

$$x_{N+1-i} = 1 - x_i \qquad (6.2\text{-}6)$$

where N is the number of grid points. Thus, substituting Eq. (6.2-6) into (1.2-3a) and (1.2-3b) yields

$$A_{ij} = \frac{1}{x_{N+1-j} - x_{N+1-i}} \prod_{\substack{k \neq N+1-i, N+1-j \\ i \neq j}}^{N} \frac{x_{N+1-i} - x_k}{x_{N+1-j} - x_k}, \qquad i, j = 1, 2, \dots, N \qquad (6.2\text{-}7)$$

and

$$A_{ii} = \sum_{k \neq N+1-i}^{N} \frac{1}{x_{N+1-i} - x_k}, \qquad i = 1, 2, \dots, N. \qquad (6.2\text{-}8)$$

So

$$A_{ij} = -A_{N+1-i, N+1-j} \qquad (6.2\text{-}9)$$

According to the definition 6.2.1 given by Eq. (6.2-1), the weighting coefficient matrix for the 1st order derivative can be concluded to be skew centrosymmetric if the grid spacing is symmetric. Next, consider the coefficient matrix for the high order derivative. Using Eqs. (6.2-8) and (6.2-9) and the recursion formulas (1.2-4) and (1.2-5) and after some manipulations, it can be shown that

$$w_{ij}^{(m+1)} = (-1)^{(m+1)} w_{N+1-i,\, N+1-j}^{(m+1)}, \qquad (6.2\text{-}10)$$

where the $w_{ij}^{(m+1)}$ denotes the DQ coefficient for the (m+1)th order derivative. Therefore, the DQ weighting coefficient matrices are skew centrosymmetric for odd order derivatives and centrosymmetric for even order derivatives when the grid spacing is symmetric.

6.2.2. Centrosymmetric structure of weighting coefficient matrices in the HDQ method

The formula (1.2-1) for determining the HDQ weighting coefficients can be restated in matrix form as

$$H w^{(m)} = H^{(m)} \qquad (6.2\text{-}11)$$

where matrix $w^{(m)}$ denotes N×N weighting coefficient matrix. m is the order of the derivative. matrix H is determined by the harmonic test functions (1.2-6) and the used grid spacing. Matrix $H^{(m)}$ depends on the test functions, the grid spacing and the derivatives order m. Premultiplying Eq. (6.2-11) by H^T yields

$$H^T H w^{(m)} = H^T H^{(m)} \qquad (6.2\text{-}12)$$

If we choose symmetric grid spacings as shown in Eq. (6.2-6), it can easily proven that the product $H^T H$ is centrosymmetric. $H^T H^{(m)}$ is centrosymmetric if m is even number or skew centrosymmetric if m is odd number. From Eq. (6.2-12), we have

$$w^{(m)} = (H^T H)^{-1} H^T H^{(m)} \qquad (6.2\text{-}13)$$

According to Lemmas 6.2.1 and 6.2.5 in the following discussions, $(H^T H)^{-1} \in C_{N\times N}$. Thus, the HDQ weighting coefficient matrices $w^{(m)}$ are centrosymmetric for even order derivatives and skew centrosymmetric for odd order derivatives.

6.2.3. Some properties of centrosymmetric and skew centrosymmetric matrices

1. Centrosymmetric matrices

For completeness, the interesting structure of the centrosymmetric matrix of (Good, 1970; Datta and Morgera, 1989) is briefly described here. $C_{N\times N}$ denotes the set of N×N centrosymmetric matrices.

Definition 6.2.3. A N dimensional vector γ is defined to be symmetric if

$$J\gamma = \gamma \qquad (6.2\text{-}14)$$

or skew symmetric if

$$J\gamma = -\gamma \qquad (6.2\text{-}15)$$

Lemma 6.2.1. If $Q_1, Q_2 \in C_{N\times N}$ and Q_1^{-1} exists, then

(1) $Q=Q_1Q_2\in C_{N\times N}$

(2) $Q=Q_1+Q_2\in C_{N\times N}$ (6.2-16)

(3) $Q_1^{-1}\in C_{N\times N}$

Lemma 6..2.2. If $Q\in C_{N\times N}$ and N=2M, Q can be written as

$$Q = \begin{bmatrix} A & JCJ \\ C & JAJ \end{bmatrix} \qquad (6.2\text{-}17)$$

where A and C are arbitrary M×M matrices. The determinant of matrix Q can be evaluated by

$$|Q| = |A + JC||A - JC| \qquad (6.2\text{-}18)$$

and the inverse of the matrix is given as

$$Q = \begin{bmatrix} P & JRJ \\ R & JPJ \end{bmatrix} \qquad (6.2\text{-}19)$$

where $2P=(A+JC)^{-1}+(A-JC)^{-1}$ and $2R=(A+JC)^{-1} - (A-JC)^{-1}$.

Applying Eqs. (6.2-18) and (6.2-19), the calculation effort of the inverse and determinant of a centrosymmetric matrix can be reduced by 75%. Note that premultiplying or postmultiplying any matrix by J only moves the elements of the matrix and requires a very little computational time.

Lemma 6.2.3. If $Q\in C_{N\times N}$ and N=2M, then

$$Q = \begin{bmatrix} A & JCJ \\ C & JAJ \end{bmatrix} \quad and \quad \begin{bmatrix} A - JC & 0 \\ 0 & A + JC \end{bmatrix} \qquad (6.2\text{-}20)$$

are orthogonally similar. Thus, the evaluation of the eigenvectors and eigenvalues of Q is equivalent to that of two M×M matrices A-JC and A+JC.

If N is odd (N=2M+1), any matrix $Q \in C_{N \times N}$ can be written as

$$Q = \begin{bmatrix} A & Js & JCJ \\ t & q & tJ \\ C & s & JAJ \end{bmatrix} \tag{6.2-21}$$

where A and C are arbitrary M×M matrices, s and t are vectors of M×1 dimension, and q is a scalar. It is easily proved that

$$Q = \begin{bmatrix} A & Js & JCJ \\ t & q & tJ \\ C & s & JAJ \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} A - JC & 0 & 0 \\ 0 & q & 2t \\ 0 & 2s & A + JC \end{bmatrix} \tag{6.2-22}$$

are orthogonally similar. Thus, the evaluation of eigenvectors and eigenvalues of Q is equivalent to those of both M×M and (M+1)×(M+1)matrices.

Lemma 6.2.4. The eigenvectors of centrosymmetric matrix Q in Lemma 3 are the sum of a symmetric vector and a skew-symmetric vector. If N is even, the N/2 skew symmetric orthonormal eigenvectors U_i and the corresponding eigenvalues λ_i can be determined by

$$(A - JC)v_i = \lambda v_i \tag{6.2-23}$$

and

$$U_i = \frac{1}{\sqrt{2}}\begin{bmatrix} v_i^T & -(Jv_i)^T \end{bmatrix}^T \tag{6.2-24}$$

The N/2 symmetric eigenvectors W_i and the corresponding eigenvalues P_i are found by solving the following equation:

$$\left(A + JC\right) y_i = P\, y_i \tag{6.2-25}$$

and

$$W_i = \frac{1}{\sqrt{2}}\begin{bmatrix} y_i^T & (Jy_i)^T \end{bmatrix}^T \tag{6.2-26}$$

If Q has distinct eigenvalues and N is odd number (2M+1), the M skew symmetric orthonormal eigenvectors U_i and corresponding eigenvalues can be determined by

$$\left(A - JC\right) x_i = \lambda_i x_i , \tag{6.2-27}$$

where $U_i = [x_i^T, 0, -(Jx_i^T)^T]^T$. The M+1 symmetric eigenvectors V_i and corresponding eigenvalues ρ_i are obtained by solving the following equation

$$\begin{bmatrix} q & 2t \\ 2s & A + JC \end{bmatrix} \begin{Bmatrix} \alpha_i \\ y_i \end{Bmatrix} = \rho_i \begin{Bmatrix} \alpha_i \\ y_i \end{Bmatrix}, \tag{6.2-28}$$

where $V_i = [y_i^T, 2\alpha_i, (Jy_i^T)^T]^T$.

2 Skew Centrosymmetric Matrices

Next, the structural properties of skew centrosymmetric matrix are discussed. The proofs for these properties are similar to those for centrosymmetric matrix (Good, 1970; Datta and Morgera, 1989), so they are omitted here for brevity. We analysis only the even order skew centrosymmetric matrix, but it should be pointed out that there are similar properties for odd order skew centrosymmetric matrix. In the following, $NC_{N\times N}$ is the set of N×N skew centrosymmetric matrices.

Lemma 6.2.5. If Q_1, $Q_2\in NC_{N\times N}$ and Q_1^{-1} exists, then
1) $Q=Q_1Q_2\in C_{N\times N}$
(2) $Q=Q_1+Q_2\in NC_{N\times N}$ (6.2-29)
(3) $Q_1^{-1}\in NC_{N\times N}$

Lemma 6.2.6. If $Q \in NC_{N\times N}$ and N=2M, then Q can be stated as

$$Q=\begin{bmatrix} A & -JCJ \\ C & -JAJ \end{bmatrix} \tag{6.2-30}$$

where A and C are arbitrary M×M matrices. The determinant of matrix Q can be expressed as

$$|Q|=|A+JC||A-JC| \tag{6.2-31}$$

and the inversion of the matrix is given as

$$Q=\begin{bmatrix} P & -JRJ \\ R & -JPJ \end{bmatrix} \tag{6.2-32}$$

where $2P=(A+JC)^{-1}+(A-JC)^{-1}$ and $2R=(A+JC)^{-1}-(A-JC)^{-1}$.

Lemma 6.2.7. If $Q\in NC_{N\times N}$ and N=2M, then

$$Q=\begin{bmatrix} A & -JCJ \\ C & -JAJ \end{bmatrix} \quad and \quad \begin{bmatrix} 0 & A-JC \\ A+JC & 0 \end{bmatrix} \tag{6.2-33}$$

are orthogonally similar. Thus, the evaluation of the eigenvectors and eigenvalues of Q is equivalent to that of a M×M matrices.

If $Q\in C_{N\times N}$ and N=2M+1, Q can be restated as

$$Q=\begin{bmatrix} A & -Js & -JCJ \\ t & 0 & -tJ \\ C & s & JAJ \end{bmatrix} \tag{6.2-34}$$

where A and C are arbitrary M×M matrices, s and t are vectors of M×1 vector. We can prove that

$$Q=\begin{bmatrix} A & -Js & -JCJ \\ t & 0 & -tJ \\ C & s & JAJ \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & -2s & A-JC \\ 2t & 0 & 0 \\ A+JC & 0 & 0 \end{bmatrix} \tag{6.2-35}$$

are orthogonally similar. Thus, the evaluation of eigenvectors and eigenvalues of Q is equivalent to that of a (M+1)×(M+1) matrix.

As can be observed from the foregoing Lemmas. the calculation of the eigenvalues and eigenvectors of a centrosymmetric matrix be reduced to those of two smaller matrices, which expedites the computational effort by nearly 75%. For skew centrosymmetric matrices, we find that there are similar computational reduction effects for the determinant, inverse, eigenvalues and eigenvectors.

We also give the following lemma on the Kronecker product involving the centrosymmetric and skew centrosymmetric matrices, e.g.

Lemma 6.2.8. If A_1, $A_2 \in V_{N\times N}$ and B_1, $B_2 \in NV_{N\times N}$, then

(1) $A_1 \otimes A_2 \in V_{N\times N}$

(2) $B_1 \otimes B_2 \in V_{N\times N}$ (6.2-36)

(3) $A_1 \otimes B_1 \in NV_{N\times N}$

The proof is straightforward and, thus, is omitted for the sake of brevity.

6.3. Applications

In this section several examples are solved by the DQ and HDQ methods, and the aforementioned centrosymmetric properties of the weighting coefficient matrices are applied to reduce the computational effort and storage requirements.

Wang and Bert's DQU approach is used in the DQ and HDQ analysis of structural components, the weighting coefficient matrices are modified in terms of the specific boundary conditions. Such coefficient matrices are also centrosymmetric when the boundary conditions are symmetric. For example, Wang and Bert (1993a) gave a detailed description for a simply supported beam. When the symmetric grid spacing is used, obviously, the modified 1st order derivative weighting coefficient matrix $[\overline{A}]$ (Eq. (3) of Wang and Bert (1993a)) is a skew centrosymmetric matrix like [A] (Eq. (2) of Wang and Bert (1993a)). The modified weighting coefficient matrix for the 2nd order derivative is given by (Eq. (4) of the same reference)

$$[\overline{B}] = [A][\overline{A}] \tag{6.3-1}$$

According to Lemma 6.2.5, $[\overline{B}]$ is a centrosymmetric matrix. The modified weighting coefficient matrix for the 4th order derivative $[\overline{D}]$ is decided by (Eq. (6) of Wang and Bert (1993a))

$$[\overline{D}] = [\overline{B}][\overline{B}] \tag{6.3-2}$$

According to Lemma 6.2.1, $[\overline{D}]$ is also a centrosymmetric matrix.

Example 1: **Flexural Vibration of a Simply Supported Beam**

The governing differential equation for this example can be expressed as:

$$W^{iv} = \varpi^2 W(x) \tag{6.3-3a}$$

where the nondimensionalized frequency is $\varpi^2 = \rho A_0 L^4 \omega^2 / EI$. A_0, L and ρ are the constant cross-sectional area, the length of the beam, the density and I the constant area moment of inertia about the neutral axis, respectively. The boundary conditions at the simply supported ends are given by:

$$w(0) = 0, \quad w''(0) = 0 \tag{6.3-3b}$$

$$w(1) = 0, \quad w''(1) = 0, \tag{6.3-3c}$$

In terms of DQU, Eq. (6.3-3a) is expressed as:

$$\sum_{j=2}^{N-1} \overline{D}_{ij} W_j = \varpi^2 W_i, \qquad i = 2, \ldots, (N-1). \tag{6.3-4}$$

Note that the boundary conditions have been applied in the formulation of the weighting coefficient matrix $\overline{D}_{ij}$, which is a centrosymmetric matrix as shown in Eq. (6.3-2) and can be written in the matrix form as:

$$\begin{bmatrix} \overline{D}_{22} & \overline{D}_{23} & \cdots & \overline{D}_{2,N-1} \\ \overline{D}_{32} & \overline{D}_{33} & \cdots & \overline{D}_{3,N-1} \\ \vdots & \vdots & \vdots & \vdots \\ \overline{D}_{N-1,2} & \overline{D}_{N-1,3} & \cdots & \overline{D}_{N-1,N-1} \end{bmatrix} \begin{bmatrix} w_2 \\ w_3 \\ \vdots \\ w_{N-1} \end{bmatrix} = \varpi^2 \begin{bmatrix} w_2 \\ w_3 \\ \vdots \\ w_{N-1} \end{bmatrix} \tag{6.3-5}$$

According to Lemma 6.2.2, we have

$$\left\{\overline{D}_{ij}\right\}_{(N-2)\times(N-2)} = \begin{bmatrix} P & JRJ \\ R & JPJ \end{bmatrix} \tag{6.3-6}$$

In the present study, eight equally spaced grid points are used. Therefore, the order of the matrices P and R is 3. In addition, it is known that the eigenvectors corresponding to the fundamental frequency of a simply supported beam is symmetric. Thus, according to lemma 6.2.4, the fundamental frequency and the corresponding eigenvector can be computed by:

$$\left(P + JR\right) u_i = \varpi^2 u_i \tag{6.3-7}$$

Similarly, the 2nd order frequency with skew-symmetric eigenvector can be calculated by:

$$\left(P - JR\right) v_i = \varpi^2 v_i \tag{6.3-8}$$

The fundamental frequency obtained by the present method is 9.8683, comparing well with the exact solution of 9.8696. The relative error is -0.006%. The 2nd order frequency is 39.2411, and the exact solution 39.4784, hence the relative error is 0.6%. If a nonuniform grid formed from the Chebyshev polynomials given in section 3.3 are used, the errors are further reduced to 0.000% for the fundamental frequency and to -0.17%for the 2nd frequency. In contrast with Wang and Bert (1993a), the present computational effort and storage requirements are reduced by 87.5% and 50%, respectively, due to the use of the centrosymmetric property of weighting coefficients matrix.

Example 2: **Transverse vibration of thin rectangular SS-SS-SS-SS and C-SS-C-SS plates**

The transverse vibration of thin rectangular SS-SS-SS-SS plate and SS-C-SS-C plates are considered in this study, where C and SS denote clamped and simply supported boundary conditions, respectively. Applying the DQ or HDQ method , the governing equation (Wang et al., 1993a, Striz et al., 1995) for this case is given by:

$$\sum_{k=2}^{N-1} \overline{D}_{ik} W_{kj} + (2\alpha^2) \sum_{m=2}^{N-1} \overline{B}_{jm} \sum_{k=2}^{N-1} \overline{B}_{ik} W_{km} + (\alpha^4) \sum_{k=2}^{N-1} \overline{D}_{jk} W_{ik} = \varpi^2 W_{ij} \tag{6.3-9}$$

where $\varpi^2=\rho h a^4 \omega^2/D$, and $\alpha=a/b=1.5$, and $\overline{D}$ and $\overline{B}$ are the modified weighting coefficient matrices for the fourth and second derivatives, respectively. N_x and N_y are the number of grid points in the X- and Y-directions (Cartesian coordinates).

The DQ method with a rather small number of grid points can produce very good results. In this case, 8×8 equally spaced grids are used. The DQ results for the fundamental of ϖ=32.0721 and for the 2nd frequency of ϖ=61.4449 agree well with the exact solutions of 32.0762 and 61.6850. The relative error is -0.01% and -0.39%. If the zeros of the shifted Chebyshev polynomials are used as the grid points, the DQ solutions for 1st and 2nd frequency are calculated as 32.0761 and 61.6159, and the errors are further reduced to -0.0003% and 0.11%. For the same problem, Wang and Bert (1993a) need compute a $(N-2)^2\times(N-2)^2$ order matrix eigenvalue problem. In contrast, the present approach only requires solving a $(N-2)^2/2 \times(N-2)^2/2$ matrix due to the use of the reducibility of centrosymmetric matrix. Therefore, the computational effort are only 12.5% of that of Wang and Bert (1993a).

The HDQ method employed equally spaced grid point and weighting coefficient matrices are modified by a combined use of the DQU and DQN techniques. HDQ with a rather small number of grid points can produces very good result. For square SS-SS-SS-SS plate, five grid points are used and the HDQ results for the fundamental frequency of $\varpi=\pi$ and for the 2nd frequency of $\varpi=2\pi$ are coincident with exact solutions. For SS-C-SS-C plate, we use eleven grid points and list some HDQ results in table 6-1, which are coincident with HDQ solutions (Striz et al., 1995) and very close to those of Leissa (1973), while the computational effort in this paper is reduced to only about 1.6% as much as that in Striz et al. (1995). This computational savings is achieved by using centrosymmetric properties of the HDQ weighting coefficient matrices (Lemmas 6.2.2 and 6.2.3 in section 6.2). We computes only a $(N^2 -2)/4\times(N^2-2)/4$ matrix for the present case. In contrast, references [2] need to solve a $(N^2 -2)\times(N^2-2)$ matrix. For all other examples with systemic boundary given in reference [2], we obtain the same results by this reduced HDQ method. It is known that when more grid points is used, the accuracies of HDQ results can be improved. Unlike the DQ method, the HDQ method have not limitation for the number of grid points. However, when more grid points is used, computing effort increases rapidly especially for multi-dimensional problems. Thus, the present reduced method will be significant in practice.

Table 6-1. Fundamental frequency ($\varpi^2=\rho ha^4\omega^2/D$) for SS-C-SS-C plates

a/b	0.4	2/3	1.0	1.5	2.5
HDQ	12.1408	17.3821	28.9656	56.3752	145.550
Leissa	12.1347	17.3703	28.9509	56.3481	145.484

Example 3. The transverse vibration of a thin, isotropic, skew plate vibration

This problem have been handled by the DQ method in (Wang et al. 1994). However, since its governing and boundary condition equations contain cross derivative term, its DQ formulation includes the centrosymmetric and skew centrosymmetric matrices simultaneously. We recalculate this problem to demonstrate the computing reduction for problems of this type by the centrosymmetric properties. The governing equation for this case is given by

$$\begin{aligned} & w_{,\xi\xi\xi\xi} - (4\cos\theta)\, w_{,\xi\xi\xi\eta} + 2(1+2\cos^2\theta)\, w_{,\xi\xi\eta\eta} - (4\cos\theta)\, w_{,\xi\eta\eta\eta} \\ & + w_{,\eta\eta\eta\eta} = (\rho h\,\omega^2 / D)\sin^4\theta \end{aligned} \qquad (6.3\text{-}10a)$$

where θ=skew angle. ρ density, ω circular natural frequency and D flexural rigidity. The simply supported boundary conditions are

$$\begin{aligned} & w = 0, \quad w_{,\xi\xi} - 2w_{,\xi n}\cos\theta = 0, \quad at \quad \xi = 0, a \\ & w = 0, \quad w_{,\eta\eta} - 2w_{,\xi n}\cos\theta = 0, \quad at \quad \eta = 0, b \end{aligned} \qquad (6.3\text{-}10b)$$

The clamped boundary conditions are

$$\begin{aligned} & w = 0, \quad w_{,\xi} = 0, \quad at \quad \xi = 0, a \\ & w = 0, \quad w_{,\eta} = 0, \quad at \quad \eta = 0, b \end{aligned} \qquad (6.3\text{-}10c)$$

let x=(2ζ-a)/a, y=(2η-b)/b. In terms of the matrix formulations (4.2-1) in section 4.2, the DQ formulation for the governing equation (6.3-10a) can be stated as

$$\begin{aligned} & \bar{D}_x W - (4\beta\cos\theta)\bar{C}_x W\bar{A}_y^T + 2\beta^2(1+2\cos^2\theta)\bar{B}_x w\bar{B}_y^T - . \\ & (4\beta\cos\theta)\bar{A}_x w\bar{C}_y^T + \beta w\bar{D}_y^T = (\bar{\omega}^2/16)w \end{aligned} \qquad (6.3\text{-}11)$$

Note that W is (n-2)×(n-2) matrix. Applying the Kronecker product of matrix to the above equation, we have

$$\begin{aligned} & \{[I_{n-2}] \otimes [\bar{D}_x] - (4\beta\cos\theta)[\bar{A}_y] \otimes [\bar{C}_x] + 2\beta^2(1+2\cos^2\theta)[\bar{B}_y] \otimes [\bar{B}_x] - , \\ & (4\beta\cos\theta)[\bar{C}_y] \otimes [\bar{A}_x] + \beta^4[\bar{D}_y] \otimes [I_{n-2}]\} vec\{W\} = (\bar{\omega}^2/16) vec\{W\} \end{aligned} \qquad (6.3\text{-}12)$$

where ⊗ denotes the kronecker product in matrix computation (the DQ formulation for this case in Eq. (10) of Wang et al. (1994a) was incorrectly expressed). $\left[I_{n-2}\right]$ represents a unite matrix, whose n is the number of grid points. vec(w) is the vector-function of a rectangular matrix formed by stacking the columns of matrix into one long vector. $\varpi^2=\left(\rho h\ a^4\ \omega^2/D\right)\ s\,i\,n^4\ \theta$. β and D are aspect ratio and flexural rigidity, respectively. Weighting coefficient matrices $\overline{A}$, $\overline{C}$, $\overline{D}$ and $\overline{B}$ are modified by the boundary conditions. For clamped boundary conditions, the $\overline{D}$ and $\overline{B}$ are the same as the rectangular plate. But the problem is more complex for simply supported boundary conditions because it contain cross derivative term $w_{,xy}$. Using the DQ method, we have $w_{,xy} = A_x\ w\ A_y^T = (A_y \otimes A_x)vec(w)$. According to Lemma 6.2.8, $A_x \otimes A_y$ is a centrosymmetric matrix since A_x and A_y are the skew centrosymmetric matrices in using symmetric distributed grid points as in Wang et al. (1994a). Therefore, $\overline{D}$ and $\overline{B}$ are centrosymmetric matrices and $\overline{A}$, $\overline{C}$ the skew centrosymmetric matrix. Obviously, the resulting coefficient matrix in the formulation equation (6.3-12) has centrosymmetric structure. Thus, we can apply the computational reduction of centrosymmetric matrix for this case similar to example 2. Namely, the computational effort is reduced by 87.5%.

Example 4. ON ANISOTROPIC PLATES

The equation governing the behaviors of mid-plane symmetric laminated plates was given by Bert et al. (1993)

$$\begin{aligned}\overline{D}_{11}w_{,xxxx} + 4\overline{D}_{16}w_{,xxxy} + 2\left(\overline{D}_{12}+\overline{D}_{66}\right)w_{,xxyy} + 4\overline{D}_{26}w_{,xyyy} + \overline{D}_{22}w_{,yyyy} \\ = q + \rho h\omega^2 w - N_x w_{,xx} - N_y w_{,yy}\end{aligned}, \qquad (6.3\text{-}13)$$

where $\overline{D}_{ij}$ are the plate stiffness, h is the total plate thickness, ρ is the density, w is the model deflections, q is the pressure only for deflection analysis, ω is the natural frequency only for free vibration analysis, N_x and N_y are uniform compression in-plane loads in x- and y- directions for buckling analysis.

In terms of the present DQ or HDQ approximate formulas (4.2-1) (Chen et al., 1996d), we have

$$\begin{aligned}\overline{D}_{11}\overline{D}_x\hat{w} + 4\overline{D}_{16}\beta\overline{C}_x\hat{w}\overline{A}_y^T + 2\beta^2\left(\overline{D}_{12}+\overline{D}_{66}\right)\overline{B}_x\hat{w}\overline{B}_y^T + 4\overline{D}_{26}\beta^3\overline{A}_x\hat{w}\overline{C}_y^T \\ + \overline{D}_{22}\beta^4\hat{w}\overline{D}_y^T = qa^4 + \varpi^2\hat{w} - \overline{N}a^2\left(\overline{B}_x\hat{w}+\hat{w}\overline{B}_y^T\right)\end{aligned}, \qquad (6.3\text{-}14)$$

where β=a/b denotes the aspect ratio, $\overline{N} = N_x = N_y$, $\varpi^2 = \rho h a^4\omega^2$. The relative boundary conditions have been taken into account in the formulation of weighting coefficient matrices by using the DQU technique, no additional equations are more required.

Applying Lemma 4.2.1 and relative corollaries, the above equation can be converted into

$$\begin{aligned}\Big[\overline{D}_{11}\left(\overline{D}_x \otimes I_y\right) + 4\overline{D}_{16}\beta\left(\overline{C}_x \otimes \overline{A}_y\right) + 2\beta^2\left(\overline{D}_{12} + \overline{D}_{66}\right)\left(\overline{B}_x \otimes \overline{B}_y\right) + \\ 4\overline{D}_{26}\beta^3\left(\overline{A}_x \otimes \overline{C}_y\right) + \overline{D}_{22}\beta^4\left(I_x \otimes \overline{D}_y\right)\Big]\vec{w} = qa^4 + \varpi^2\vec{w} -. \qquad (6.3\text{-}15) \\ \overline{N}a^2\left(\overline{B}_x \otimes I_y + I_x \otimes \overline{B}_y\right)\vec{w}\end{aligned}$$

The above formulation equation is equivalent to equation (13) of Bert et al. (1993, 1994a). The present procedures obviously simplify formulation effort and are much more easy for programming, and the resulting formulation has a explicit matrix form.

For problems with symmetric boundary conditions such as the simply supported or clamped anisotropic plates as discussed in Bert et al. (1993, 1994a), it is straightforward that $\overline{A}_x$, $\overline{A}_y$, $\overline{C}_x$ and $\overline{C}_y$ are skew centrosymmetric matrix and $\overline{B}_x$, $\overline{B}_y$, $\overline{D}_x$ and $\overline{D}_y$ are centrosymmetric matrix when the uniform gird points or the zeros of the Chebyshev polynomials are used. According to Lemmas 6.2.2 and 6.2.8, the resulting coefficient matrix in the formulation equation (6.3-15) is a centrosymmetric matrix. Therefore, the reduction algorithm based on the factorization properties of centrosymmetric matrix (Chen et al., 1996a, b) is applicable for the cases, namely, the computational effort is reduced to 75 per cent as much as that in Bert et al. (1993, 1994a).

Bert et al. (1993, 1994a) pointed out that the DQ and HDQ methods were very competitive technique to analyze static and dynamic behaviors of the anisotropic plates. The present work makes the methods more computationally efficient and easier to be used for these problems. But it is regret that the reduced method is only available for problems with systemic boundary conditions in the DQU approach. For the deflection and vibrational analysis of plates and beams with non-systemic boundary conditions, the present reduction approach is also applicable, but the DQU approach can be not used for the cases. The DQN and DQZ approaches discussed in section 4.4 should be used, in which the governing and boundary condition equations can be handled separately.

Example 5. Convection-diffusion problem

Civan and Sliepcevich (1984b) computed the convection-diffusion problems by the differential quadrature method. We analyze this example again to show that the computational reduction in centrosymmetric matrix can be used for those problems with non-symmetric boundary conditions. We consider the steady-state case. In terms of the DQ method, the governing equation (Civan and Sliepcevich, 1984b) can be expressed as

$$-\frac{\phi_{ij}}{4a} + \alpha\sum_{k=1}^{N^x} b_{ik}^{\,x}\ \phi_{kj} + \beta\sum_{k=1}^{N^y} b_{jk}^{\,y}\ \phi_{jk} = 0, \tag{6.3-16a}$$

where α and β are constant quantities. b_{ik}'s are the DQ weighting coefficient for the second derivative and have been not modified by boundary conditions, which is different from examples 1-3. The boundary conditions are approximated by the DQ method as

ϕ_{ij}=prescribed value, j=1,2, ..., N^y (6.3-16b)

ϕ_{iN^y} = 0, j=1,2, ..., N^x (6.3-16c)

$$\frac{\partial\phi_{ij}}{\partial y} = \sum_{k=1}^{N^y} a_{jk}^{\,y}\ \phi_{ik} = 0, \quad \text{at } j=1. \tag{6.3-16d}$$

$\phi_{N^x j}$ =0, j=1,2, ..., N^x. (6.3-16c)

Since the boundary conditions in this problem are not systemic, we need handle with Eq. (6.3-16a) and Eqs. (6.3-16b,c) separately. Obviously, the resulting coefficient matrix in Eq. (6.3-16a) is a centrosymmetric matrix irrespective of whether the boundary conditions are symmetric or non-symmetric. Solving the Eq. (6.3-16a) by using the centrosymmetric properties, the function values at inner grid points are expressed by the side lateral function values using a matrix formula. Substituting this formula into the boundary condition equations (6.3-16b, c), we can obtain the algebraic equations only containing the side lateral function values. The order of the equations is much smaller than that of Eq. (6.3-16a). The computational effort for solving it will be little. The main computational effort in this procedure is solving

Eq. (6.3-16a), while the computational reduction by using the centrosymmetric matrix can be achieved in this step. So the present computational effort is nearly only 25% as that in Civan and Sliepcevich (1984b).

In section 4.3 we discussed the reducibility in computing the Poisson equation and convection-diffusion equations by means of the fast algorithms in the solution of the Lyapunov equation. In th0se cases with symmetric boundary conditions, we can factorize the respective centrosymmetric coefficient matrices $\overline{B}_x$ and $\overline{B}_y$ into two smaller size sub-matrices, nearly half, in all the four computing steps of the BS, HS and R-THR methods. Therefore, the computing effort can be further reduced to 8.5% under 7×7 grid points and 1.5% under 11×11 grid points as that in Civan and Sliepcevich (1983b; 1984b).

6.4. Nonlinear Computations and Centrosymmetric Matrix

Section 6.3 applies the factorization properties of the centrosymmetric and skew centrosymmetric structures to reduce computing effort and storage requirements in the DQ solution of the linear problems by 75% and 50%, respectively. For nonlinear differential equations, the centrosymmetric structure of the DQ weighting coefficient matrices can be also exploited to reduce the computational effort by using the Hadamard product and SJT product concept. We present the following lemmas on the Hadamard product and SJT product involving the centrosymmetric and skew centrosymmetric matrices. The proofs for them are straightforward and thus omitted for brevity.

Lemma 6.4.1. If A_1, $A_2 \in V_{N\times N}$ and B_1, $B_2 \in NV_{N\times N}$, then
(1) $A_1 \circ A_2 \in V_{N\times N}$
(2) $B_1 \circ B_2 \in V_{N\times N}$
(3) $A_1 \circ B_1 \in NV_{N\times N}$

Lemma 6.4.2. If $A_1 \in V_{N\times N}$, $B_1 \in NV_{N\times N}$ and $P_{1,}$ $P_2 \in C^{N\times 1}$, P_1 is symmetric P_2 is skew symmetric, then
(1) $A_1 \Diamond P_1 \in V_{N\times N}$
(2) $B_1 \Diamond P_1 \in NV_{N\times N}$
(3) $A_1 \Diamond P_2 \in NV_{N\times N}$
(4) $B_1 \Diamond P_2 \in V_{N\times N}$

According to Lemmas 6.4.1 and 6.4.2, we can find that the computational reduction by using the factorization properties of centrosymmetric matrix can be easily extended to matrix computations involved the Hadamard product and SJT product. We recalculate equations (2.8-15) and (2.5-1a, b) by the DQ method to demonstrate the computational reduction by using the factorization properties of centrosymmetric matrices when the symmetric grid spacings such as the equally spaced grid points and the roots of the shifted Legender or the shifted Chebyshev polynomials, are employed. For the DQ solution of geometrically nonlinear bending of the orthotropic rectangular plates with symmetric boundary conditions, the computing effort and storage requirements can be further reduced by 75% and 50%, respectively, using such reduction technique. The details on these applications are straightforward and not presented for brevity. It should be pointed out that the reduction technique is in general applicable for the nonlinear problems with symmetric structures.

6.5. Conclusions

The study in this chapter proves in general the centrosymmetric structures of the weighting coefficient matrices of the DQ and HDQ methods when the grid spacing is symmetric. Also a new type of matrix, called as the skew centrosymmetric matrix, is defined and its main properties are discussed. The work furthers the knowledge and understanding of the DQ and HDQ methods. The structure of such matrices allows for factorization of the determinant and the characteristic equations into two, smaller and nearly equal size matrices. Therefore, the computational complexity can be reduced by 75%. Furthermore, it is shown that these matrices possess either symmetric or skew-symmetric eigenvectors. In many problems, the need of symmetric or skew-symmetric eigenvectors can be applied as a constraint. Therefore, the computational effort can be reduced further. The present work emphasizes the importance of the analysis of the weighting coefficient matrices in the DQ-type methods and makes the methods even more attractive for the engineering analysis.

CHAPTER 7.

SUGGESTIONS FOR FUTURE WORK

The current situations for the research of the DQ-type methods are somewhat similar to those for the FEM in the early 60's and for the BEM in the early 70's, while the Hadamard product and SJT product are still at an early developing stage for nonlinear computation and analysis of general purpose. Several major obstacles, which have always impeded the DQ methods, have been overcome due to our and other researchers' recent work. However, some important aspects remains largely an unclear matter in the DQ method and relative applications, for example, round off error, numerical stability, and applications for problems with complex geometries. In many cases, the complexity of problems manifests itself in the degree of nonlinearity of the system, its scale, transient nature or the interaction between components of the system, while the salient merit of the DQ method is not well explored and demonstrated in solving a fairly wide range of complex problems. The differential cubature method is a new development of the differential quadrature method and show good promise for multidimensional problems, but many essential aspects of this method are not involved in the literature, and the applications are very limited. In this chapter, we hope to outline a few key problems in applying the DQ-type methods and the Hadamard product and SJT product techniques to practical engineering. Both theory and applications should be equally emphasized.

Coordinate mappings (Lam, 1993; Bert and Malik, 1996c) and multidomain approaches (Civan, 1985; Shu and Richards, 1992; Striz, Chen and Bert, 1994) have been used to maintain the high accuracy and convergence properties in the DQ method for handling problems with complex geometries. By using multidomain technique, the DQ method can effectively analyzed the problems with irregular domain, which is composed of some regular domains. Coordinate mappings approach is exploited for arbitrary irregular shaped problems. The applications are successful for systems of no higher than second order. However, the governing equations in structural mechanics problems are in general higher order differential equations. The coordinate mappings approach encounters some limitations for structural problems with simply-supported or free boundary conditions due to inaccuracies of geometric mapping. Also, it was found that the approach seems not to be suitable for the curvilinear non-quadrangular shaped problems for the same reason. It was proposed by Bert and Malik (1996c) to use more accurate mapping approach to overcome these difficulties, but the method may require more mapping effort. We have a different idea for handling this problem. For example, considering a two-dimensional problems, we suggest applying the DQ method to analogize the spatial derivatives or integral in one direction only, and then, the governing equations for this problem are transformed into a set of ordinary differential equations of boundary value problem, while the solution procedures for ordinary differential equations have been well developed. The simplicity and efficiency of the approach can be demonstrated via some simple examples. The further work is hoped. In addition, a combined use of the DQ method and other numerical techniques such as the FE, FD and BE methods is expected to be effective for problems with irregular shape, since the similar idea has been applied in the collocation, pseudo-spectral, spectral and Galerkin methods for the same purpose.

From the standpoints of engineering applications, the DQ method is still at its very first step. The extension of the technique to analyze complex problems in new realm is another important task in the future study. In recent research, a neglected issue in the DQ method has been inverse problems, which arise in such diverse areas as the study of drug distributions in the body, weather prediction, structural optimization designs, and modal parameter identifications. The DQ method has been shown to be a efficient approach in system identification problems in its early study (Bellman et al., 1974a, b, 1979; Hu et al., 1974). However, the work in this direction can not be continued due to the lack of understanding in some basic aspects of the method. Now these impedances have been overcome. The DQ method is hoped to be very efficient for handling problems of structural optimization designs and modal parameter identifications. Of course, the DQ method also has its disadvantages for certain problems. Therefore, it is necessary to employ a combination of a few methods for some complex problems so as to utilize their respective advantages. For example, a combined use of the DQ and BE methods may be useful for some problems such as elasto-plastic fracture mechanics problems.

Some basic aspects of the differential cubature method such as stability analysis, truncation error, accuracy and structures of weighting coefficients, and the choice of test functions and sampling points have not been fully studied. It is also claimed that the DC method is a superior numerical technique for multidimensional problems. However, its potential in the solution of a varied class of problems has been not sufficiently explored. It is hoped to apply the DQ and DC methods to the weather prediction problems (Chen and Zhong, 1996c). The nonlinear differential equations occurring in global and hemispheric weather prediction are traditionally computed by the finite difference method and the spectral method based on the spherical harmonics (Haltiner and Williams, 1980; Jarrand and Simmons, 1983). A combination of the DQ and DC methods may yield a potential alternative numerical modeling for such kind of problems. The spherical harmonics can be chosen as the test functions in the DC method instead of the conventional monomials to approximate partial derivatives with respect to variables along the spherical surface coordinate direction, while the partial derivative with respect to the vertical direction variable can be approximated by the DQ method. The variable domain of global and hemispheric weather prediction is regular, while the DQ and DC methods have been very successfully applied for some simpler nonlinear problems with regular geometries. Therefore, it is expected that the DQ and DC methods can also succeed in this task.

Another important aspect in applying the DQ method to engineering is nonlinear stiff initial value problems. The detailed discussions can be found in section 5.7. Bert’s group as well as other researchers has analyzed the bending, vibration, and buckling problems for a variety of structural components, which are mainly limited to regular domain. We think that the work in this field has been rather sufficient for research purpose. More work should be place on the applications of the DQ method for structural problems involving nonlinearity and irregular domain. The boundary value problems with irregular geometries and nonlinear stiff initial value systems will be key to apply the DQ method to engineering. The similarity between the DQ method and the numerical integration methods (Philip et al., 1975) can provide some new insights in the study of round off error, convergence properties, numerical stability, and inherent limitations of the DQ method. The properties of the Hadamard product may provide some innovations in a further study of stability and round off in the nonlinear computations of the DQ method. Also, since the DQ method has close relation with the method of weighted residuals (MWR), the knowledge on the MWR (Xu, 1987) can be introduce to the study of the DQ method and its applications. In addition, the semi-analytical technique (Zao, 1992) can be extended to the DQ method.

The Hadamard product and SJT product approach will provide a new framework to general nonlinear computations, modeling and analysis from the standpoints of algorithmic simplicity, numerical accuracy and computational efficiency. However, their applications are now in the beginning development stage. The efficiency and simplicity of the techniques are not sufficiently investigated in the other numerical computational methods, including the FE, FD, BE, spectral, pseudo-spectral, collocation, Rayleigh-Ritz, Galerkin, Wilson θ, Houblot, Newmark, Runge-Kutta, Gear, and some variants of these methods. Some important nonlinear problems such as nonlinear system identifications and optimization design are not involved in the existing work. Recently, the Hadamard product has been found to be a very powerful concept in several engineering areas. Song and Middleton (1992) applied the Hadamard product to construct the robot dynamics of parameter-isolated form and presented a new effective control scheme for systems with rapidly varying parameter. Diggelen and Glover (1994) introduced the Hadamard product to linear controllers design for robust decoupling. Ahn’s paper (1993) was concerned with exploiting the Hadamard product to formulate an adaptive algorithm for parameter identification problem in IIR systems. These work centered on stability problems in optimal control theory. On the other hand, structural optimum design has been a very important branch in computational mechanics and has widespread applications in practical engineering design. As was pointed out by Zhong et al. (1992), the mathematical problems in optimal control theory simulates the structural static problems. Based on the work of Zhong et al. (1992), the present authors think that the foregoing work can be extended to solve structural optimum design problems. It is worth pointing out that the Hadamard product and SJT product approach can make nonlinear computations be easily accomplished in a parallel treatment way. This is of vital importance for analysis of nonlinear problems of large scale.

What is quite important is that the numerical techniques are straightforward, easily learned, readily programmed, and easily used. We think the DQ-type methods and Hadamard product and SJT product techniques do have these merits. It is essential that new contributions in numerical methodology and associated algorithms be applied rapidly to the practical engineering and scientific areas. Therefore, it is also our hope that a further work will provide a solid basis for developing an efficient commercial software package so that the DQ-type methods and Hadamard product and SJT product techniques are firmly established as one of the most popular numerical techniques in engineering.

REFERENCES

[1] Ahn, C. (1993), “An adaptive algorithm for identification of IIR Systems”, Proc. 32nd Conf. Decision Contr. San Antonio, Texas, Dec. 2595-2600.

[2] Atkinson, K. E. (1978), An introduction to numerical analysis, John Wiley & Sons.

[3] Bartels, R. H.; Stewart, G. W. (1972), “A solution of the equation AX+XB=C”, Communications in AM **15**, 820-826.

[4] Bathe, K. J.; Wilson, E. L. (1976), Numerical Methods in Finite Element Analysis, Englewood Cliffs, New Jersy: Prentice-Hall, Inc.

[5] Bathe, K. J. (1982), Finite Element Procedures in Engineering Analysis, Prentice-Hall Inc., Enlewood Cliffs, NJ.

[6] Bazeley, G. P.; Cheung, Y. K.; Irens, B. M.; Zienkiewicz, O. C. (1965), “Triangular elements in bending-conforming and non-conforming solutions”, Proc. Conf. Matrix Methods in Structural Mechanics, J. S. Przemieniecki et al. eds. Chio: Air Force Ins. Tech. Wright-Patterson A F Base. 547-576.

[7] Bellman, R. E.; Casti, J. (1971), “Differential quadrature and long-term integration”. Journal of Mathematical Analysis and Applications **34**, 235-238.

[8] Bellman, R. E.; Kashef, B. G.; Casti, J (1972), “Differential quadrature: A technique for the rapid solution of nonlinear partial differential equations”, J. Comput. Phys.**10**, 40-52.

[9] Bellman, R. E. (1973), Methods of Nonlinear Analysis, Vol. 2, chap. 16, Academic Press, New York.

[10] Bellman, R. E.; Kashef, B. G. (1974a), “A scanning technique for systems identification”, Journal of Mathematical Analysis and Applications **71**, 403-411.

[11] Bellman, R. E.; Kashef, B. G.; Vasudevan, R. (1974b), “The inverse problem of estimating heart parameters from cardiograms”, Math. Biosci. **19**, 221-230.

[12] Bellman, R.; Kashef, B.; Lee, E. S.; Vasudevan, R. (1975), “Differential quadrature and spline”, Comput. & Math. with Appl. **1**, 371-376.

[13] Bellman, R.; Roth, R. S. (1979), “System identification with partial differential”, J. Math. Anal. Appl., **68**, 321-333.

[14] Bellman, R; Adomian, G. (1985), Partial differential equations, D Reidel Publishing, Dordrecht, The Netherlands.

[15] Bellman, R; Roth, R. S. (1986), Methods in approximation, D Reidel Publishing, Dordrecht, Netherlands.

[16] Bert, C. W.; Jang S. K.; Striz, A. G. (1988a), “Two new methods for analyzing free vibration of structural components”, AIAA Journal, **26**, 612-618.

[17] Bert, C. W.; Cho, K. N. (1988b), Private communication. The University of Oklahoma.

[18] Bert, C. W.; Striz, A. G.; Jang, S. K. (1989), “Nonlinear bending analysis of orthotropic rectangular plates by the method of differential quadrature”, Comput. Mech. **5**, 217-226.

[19] Bert, C. W.; Wang, X.; Striz, A. G. (1993), “Differential quadrature for static and free vibrational analysis of anisotropic plates”, Int. J. Solids Structures **30**, 1737-1744.

[20] Bert, C. W.; Wang, X.; Striz, A. G. (1994a), “Convergence of the DQ method in the analysis of anisotropic plates”, J. Sound & Vibr. **170**, 140-144.

[21] Bert, C. W., Wang, X, Striz, A. G. (1994b), “Static and free vibrational analysis of beams and plates by differential quadrature method,” Acta Mech. **102**, 11-24.

[22] Bert, C. W.; Malik, M. (1996a), “Free vibration analysis of thin cylindrical shells by the differential quadrature method”, ASME J. Pressure Vessel Tech. **118**, 1-12.

[23] Bert, C. W.; Malik, M. (1996b), “Free vibration analysis of tapered rectangular plates by differential quadrature method: A semi-analytical approach”, J. Sound & Vib. **190**, 41-63.

[24] Bert, C. W.; Malik, M. (1996c), “The differential quadrature method for irregular domains and applications to plate vibration”, Int. J. Mech. Sci., **38**, 589-606.

[25] Bert, C. W.; Malik, M. (1996d), “Differential quadrature method in computational mechanics: A review”, Appl. Mech. Rev. **49**, 1-28.

[26] Bjorck, A. A.; Pereyra, V. (1970), “Solution of Vandermonde system of equation”, Math. Comput. **24**, 893-903.

[27] Burden, R. L.; Faires, J. D.; Reynolds, A. C. (1978), Numerical Analysis, 2nd Eds., Prindle, Weber & Schmilt, Boston.

[28] Burka, M. K. (1982) “Solution of stiff ordinary differential equations by decomposition and orthogonal collocation”, AIChE Journal **28**, 11-20.

[29] Cantoni, A and Butler, P (1976), “Eigenvalues and eigenvectors of symmetric centrosymmetric matrices”, Linear Algebra & its Applications, **13**, 275-288.

[30] Canuto, C., Hussaini, M. Y.; Quarteroni, A., Zang, T. A. (1988), Spectral Methods in Fluid Dynamics, Springer, Berlin.

[31] Cao, Zhiyuan (1989), Vibration Theory of Plates and Shells (in Chinese), Chinese Railway Press, Beijing.

[32] Cao, Zhiyuan and Cheung, Y. (1992), Semi-analytical Numerical Methods (in Chinese), Defense Industry Press, Beijing.

[33] Chang, C. T.; Tsai, C. S.; Lin, T. T. (1993), “Modified differential quadratures and their applications”, Chem. Eng. Commun. **123**, 135-164.

[34] Chen, Wen; Yu, Yongxi (1993a), “Calculation and analysis of weighting coefficient matrices in differential quadrature”, Proc. of the 1st Pan-Pacific Conference on Computational Engineering, Seoul, Korea, 1-5 Nov., 1993, Elsevier Science Publ. B. V. , Netherlands, 157-162.

[35] Chen, Wen; Yu, Yongxi (1993b), “Differential quadrature method for high order boundary value problems”, Proc. of the 1st Pan-Pacific Conference on Computational Engineering, Seoul, Korea, 1-5 Nov., 1993, Elsevier Science Publ. B. V. , Netherlands, 163-168.

[36] Chen, Wen (1994), “Differential quadrature method and its applications to structural engineering”, MS thesis, Shanghai Jiao Tong University.

[37] Chen, Wen; Wang X.; Zhong, Tingxiu. (1996a), “The structure of weighting coefficient matrices of harmonic differential quadrature method and its applications”, Comm. Numer. Method Eng. **12**, 455-461.

[38] Chen, Wen; Yong, Y.; Wang, X. (1996b), “Reducing the computational effort of the differential quadrature method”, Numer. Methods for P. D. E. **12**, 565-577.

[39] Chen, Wen; Zhong, Tingxiu (1996c), “The study on nonlinear computations of the DQ and DC methods”, Numer. Methods for P. D. E. (in press).

[40] Chen, Wen; Zhong, Tingxiu; He, W. (1996d), “A Note on the DQ Analysis of Anisotropic Plates”, J. Sound & Vibration (accepted).

[41] Chen, Wen; Zhong, Tingxiu; Zhang, Jianwu (1996e), “Hadamard product and nonlinear computations -I”, Inter, J. Numer. Methods in Engrg. (to appear).

[42] Chen, Wen; Zhong, Tingxiu; Zhang, Jianwu (1996f), “Hadamard product and nonlinear computations -II”, Inter, J. Numer. Methods in Engrg. (to appear).

[43] Chen, Wen; Zhong, Tingxiu; Liang, S. (1996g), “Structural Dynamic Analysis Using Differential Quadrature Method”, J. Sound & Vibration (to appear).

[44] Chen, Wen; Zhong, Tingxiu; He, S (1996h), “The DQ Solution of Geometrically Nonlinear Bending of Orthotropic Rectangular Plates by Using Hadamard and SJT Product”, Computers & Structures (to appear).

[45] Chen, Wen; Zhong, Tingxiu (1996i), “Minimized Computational Effort in Applying the DQ Method to the Poisson and Convection-Diffusion Equations”, J. Comput. Phys. (to appear).

[46] Chen, Wen; Liang, S.; Zhong, Tingxiu (1996j), “On the DQ Analysis of Geometrically Nonlinear Vibration of Immovably Simply Supported Beams”, J. Sound & Vibration (to appear).

[47] Cheng, Y. P. (1987), Matrix theory, West-North Univ. Tech. Press, Xian.

[48] Civan, F.; Sliepcevich, C. M. (1983a), “Application of differential quadrature to transport processes,“ J. Math. Anal. Appl. **93**, 206-221.

[49] Civan, F.; Sliepcevich, C. M. (1983b), “Solution of the Poisson equation by differential quadrature, Int. J. Numer. Methods Eng. **19**, 711-724.

[50] Civan, F.; Sliepcevich, C. M. (1984a), “On the solution of the Thomas-Fermi equation by differential quadrature,” J. Comput. Phys., **56**, 343-348.

[51] Civan F.; Sliepcevich, C. M. (1984b), “Differential quadrature for multidimensional problems”, J. Math. Anal. Appl., **101**, 423-443.

[52] Civan F.; Sliepcevich, C. M. (1985), “Application of differential quadrature to solution of pool boiling in cavities”, Proc. Okla. Acad. Sci. **65**, 73-78.

[53] Civan F.; Sliepcevich, C. M. (1986), “Solving integro-differential equations by the quadrature method”, Proc. of the 1st Int. Conference on Integral Methods in Sci. & Engrg., UT Arlington, Mar. 18-21, 1985, Hemisphere Publ. Co., New York, 106-111.

[54] Civan, F. (1989), “Differential cubature for multidimensional problems”, in Proc. of the 20th Annual Pittsburgh Conference on Modeling and Simulation, W. G. Vogt and M. H. Mickle, Eds., Instrument Society of American, North Carolina, p. 1843-1847.

[55] Civan, F. (1993), “Comments on application of generalized differential quadrature to solve two-dimensional incompressible Navier-Stokes equations by C. Shu and B. E. Richards”, Int. J. Numer. Methods Fluids, **17**, 921-922.

[56] Civan, F. (1994a), “Solving multivariable mathematical models by the quadrature and cubature methods”, Numer. Methods for P. D. E., **10**, 545-567.

[57] Civan, F. (1994b), “Rapid and accurate solutions of reactor models by the quadrature method”, Computers & Chemical Engineering, **18**, 10005-10009.

[58] Cook, R. D. (1974), Concepts and Applications of Finite Element Analysis, John Wiley & Sons, Inc.

[59] Dahlquist, G. (1963), “A special stability problem for linear multistep methods”, **BIT**, 3, 27-43.

[60] Datta, L.; Morgera, S. D. (1989), “On the reducibility of centrosymmetric matrices applications in engineering problems”, Circuits Systems Signal Process, **8**, 71-96.

[61] Diggelen, F. V.; Glover, K. (1994), “A Hadamard weighted loop shaping design procedure for robust decoupling”, Automatica, **30**, 831-845.

[62] Dodd, R. K.; Eilbeck, J. C.; Gibbon, J. D.; Morris, H. C. (1982), Solitions and nonlinear wave equations, Academic Press, Orlando.

[63] Dokainish, M. A.; Subbaraj, K.(1989a), “A survey of direct time-integration methods in computational structural dynamics-I”, Computer & Structures **32**, 1371-1386.

[64] Dokainish, M. A.; Subbaraj, K. (1989b), “A survey of direct time-integration methods in computational structural dynamics-II”, Computer & Structures **32**, 1387-1401.

[65] D’souza, F.; Garg, V. K. (1984), Advanced Dynamics - Modeling and Analysis, Chapt. 7, Prentice-Hall, Inc., New Jersey.

[66] Du. H.; Lim, M. K.; Lin, R. M. (1994), “Application of generalized differential quadrature method to structural problems”, Int. J. Numer. Eng. **37**, 1881-1896.

[67] Engles, H. (1980), Numerical quadrature and cubature, Academic Press, San Diego.

[68] Farsa, J. Kukreti A. R.; Bert, C. W. (1991), “Fundamental frequency of laminated rectangular plates by differential quadrature method”, Comput. Struct. **46**, 465-477.

[69] Feng, Y.; Bert, C. W. (1992), “Application of the quadrature to flexural vibration analysis of a geometrically nonlinear beam”, Nonlinear Dynamics, **3**, 13-18.

[70] Finlayson, B. A. (1972), The Method of Weighted Residuals and Variational Principles, Academic Press, New York.

[71] Finlayson, B. A. (1980), Nonlinear Analysis in Chemical Engineering, McGraw-Hill, New York.

[72] Gear, C. W. (1967), “The numerical integration of ODE”, Math. Comp. 21, 146-156.

[73] Gear, C. W. (1971a), “Simultaneous numerical solution of differential -algebraic equations”, IEEE Transaction Circuit Theory **18**, 89-95.

[74] Gear, C. W. (1971b), Numerical initial value problems in ordinary differential equations, Prentice-Hall, Inc.

[75] Golub, H.; Nash, S.; Vanloan, C. (1979), “A Hessenberg-Schur method for the problem AX+XB=C”, IEEE Transactions on Automatic Control AC-**24(6)**, 909-913.

[76] Good, I. J. (1970), “The inverse of a centrosymmetric matrix,” Technometrics **12**, 925-928.

[77] Gui, Bing (1992), “A Numerical Solution of Lyapunov Matrix Equation and a Partly Parallel Treatment (in Chinese)”, Journal of Nanjing University of Aeronautics and Astronautics **24**, 449-455.

[78] Gutierrez, R. H.; Laurra, P. A. A. (1994), “Solution of the Helmholtz equation in a parallelogrammic domain with mixed boundary conditions using the differential quadrature method”, J. Sound & Vib. **178**, 269-271.

[79] Haltiner, G. J.; Williams, R. T. (1980), Numerical weather prediction and dynamic meteorology, 2nd edition, John Wiley & Sons, Inc., New York.

[80] Hamming, R. W. (1973), Numerical methods for scientists and engineers, McGraw-Hill, New York.

[81] Holden, J. T. (1972), “On the finite deflections of thin beams”, Int. J. Solids Struct. **8**, 1051-1055.

[82] Horn, R. A.; Johnson, C. R. (1985), Matrix Analysis, Cambridge Univ. Press.

[83] Horn, R. A. (1990), “The Hadamard product”, in Matrix theory and applications, Series: Proc. of Symposia in applied mathematics, V. 40, C. R. Johnson Ed., American Mathematical Society, Providence, Rhode Island, , p. 87-169.

[84] Hu, L. C.; Hu, C. (1974), “Identification of rate constants by differential quadrature in partly measurable compartmental models,” Math. Biosci. **21**, 71-76.

[85] Huebner, K. H. (1975), The Finite Element Method for Engineers, John Wiley & Sons Inc., New York.

[86] Jang, S. K.; Bert, C. W.; Striz, A. G. (1989), “Application of differential quadrature to static analysis of structural components”, Inter. J. Numer. Methods Eng., **28**, 561-577.

[87] Jarrand, M.; Simmons, A. J. (1983), “The spectral techniques. Numerical methods for weather prediction”, ECMWF Seminar, Vol. **2**, 1-59.

[88] Kang, K.; Bert, C. W.; Striz, A. G. (1995), “Vibration analysis of shear deformable circular arches by the differential quadrature method”, J. Sound & Vibr. **183**, 353-360.

[89] Kang, K; Bert, C. W.; Striz, A. G. (1996), “Static analysis of a curved shaft subjected to end torques”, Int. J. Solids and Structures, **33**, 1587-1596.

[90] Kashef, B. G.; Bellman, R. E.; (1974), “Solution of the partial differential equation of the Hodgkins-Huxley model using differential quadrature”, Math. Biosci. **19**, 1-8.

[91] Kukretic, A. R. Farsa, J. Bert, C. W. (1992), “Fundamental frequency of tapered plates by differential quadrature”, J. Eng. Mech. **118**, 1221-1238.

[92] Lam, S. S. E. (1993), “Application of the differential quadrature method to two-dimensional problems with arbitrary geometry”, Computers & Structures, **47**, 459-464.

[93] Lancaster, P.; Timenetsky, M. (1985), The Theory of Matrices with Applications, 2nd edition, Academic Press. Orlando.

[94] Lapidus, L.; Pinder, G. F. (1982), Numerical Solution of Partial Differential Equations in Science and Engineering, John Wiley & Sons Inc., New York.

[95] Laurra, P. A.; Saffell, B. F. Jr. (1967), “Small-amplitude vibration of clamped rectangular plates using polynomial approximations”, J. of Acoustic Society of America, **41**, 836-839.

[96] Laurra, P. A. A.; Gutierrez, R. H. (1993), “Analysis of vibrating Timoshenko beams using the method of differential quadrature”, Shock Vib. **1**, 89-93.

[97] Laurra, P. A. A.; Gutierrez, R. H. (1994a), “Analysis of vibrating rectangular plates with non-uniform boundary conditions by using the differential quadrature method”, J. Sound & Vibr. **173**, 702-706.

[98] Laura, P. A. A.; Rossi, R. E. (1994b), “The method of differential quadrature and its application to the approximate solution of ocean engineering problems”, Ocean Eng. **21**, 57-66.

[99] Laura, P. A. A.; Gutierrez, R. H.; Rossi, R. E. (1996), “Free vibrations of beams of bilinearly varying thickness”, Ocean Eng. **23**, 1-6.

[100] Leissa, A. W. (1973), “The Free Vibration of Rectangular Plates”, J. Sound & Vibr., **31**, 257-293.

[101] Levy, S. (1942a), “Square plate with clamped edges under normal pressure producing large deflections”, NACA Report No. 740.

[102] Levy, S. (1942b), “Bending of rectangular plates with large deflections”, NACA Report No. 737. 139-157.

[103] Liang S.; Li H.; Chen Wen (1997), “The DQ solution of buckling of circular arches under water loading”, J. Huazhong Univ. of Sci. & Tech., **25**, 81-82, 1997.

[104] Liew, K. M.; Han, J. K.; Xiao, Z. M.; Du, H. (1996), “Differential quadrature method for mindlin plates of Winkler foundations”, Int. J. Mech. Sci., **38**, 405-421.

[105] Lin R. M.; Lim M. K.; Du, H. (1994), “Deflection of plates with nonlinear boundary supports using generalized differential quadrature”, Comput. Struct. **53**, 993-999.

[106] Lü, H.; Zhu, J.; Ma, L. (1995), “Discussion of analyzing of geometric non-linear beams with large rotations”, Comput. Struct. Mech. Appl. **12**, 485-490.

[107] Lü, D. (1980), Numerical recipes in Fortran (in Chinese), Defense Industry Press.

[108] Malik, M; Bert, C. W.; Kukretic A. R. (1993), “Differential quadrature solution of uniformly loaded circular plate resting on elastic half-space”, in Aliabadi, M. H. and Brebbia, C. A. (eds), Contact Mechanics: Computational Technique, 385-396, Computational Mechanics Publications, Southampton, UK.

[109] Malik, M.; Civan, F. (1994a), “A comparative study of differential quadrature and cubature methods vis-à-vis some conventional techniques in context of convection-diffusion-reaction problems”, Chemical Engineering Science, **50**, 531-547.

[110] Malik, M.; Bert, C. W. (1994b), “Differential quadrature solutions of steady-state incompressible and compressible lubrication problems”, Journal of Tribology **116**, 296-302.

[111] Malik, M.; Bert, C. W. (1996), “Implementing multiple boundary conditions in the DQ solution of higher order PED’s: Application to free vibration of plates”, Int. J. Numer. Methods Eng. **39**, 1237-1258.

[112] Mansell, G.; Merryfield, W.; Shizgal, B.; Weinerl, U. (1993), “A comparison of differential quadrature methods for the solution of partial differential equations”, Computer Methods in Applied Mechanics and Engineering, **104**, 295-316.

[113] Mattiasson, K.; Bengtsson, A.; Samuelsson, A. (1986), “On the accuracy and efficiency of numerical algorithms for geometrically nonlinear structural analysis. in Finite Element Methods for Nonlinear Problems”, Proc. of the Europe-US Symposium, P. G. Bergan, K. J. Bathe, W. Wunderlich, ed., Springer-Verlag, Berlin.

[114] Mei, C. (1973), “Finite element displacement method for large amplitude free flexural vibration of beams and plates”, Computers & Structures **3**, 163-174.

[115] Mingle, J. O. (1973), “Computational considerations in nonlinear diffusion”, Inter. J. Numer. Methods Engrg. **7**, 103-116.

[116] Mingle, J. O. (1977), “The method of differential quadrature for transient nonlinear diffusion”, J. Math. Anal. Appl. **60**, 559-569.

[117] Morgera, S. D. (1982), “On the reducibility of finite Toeplitz matrices-applications in speech analysis and pattern recognition”, Signal Processing, **4**, 425-443.

[118] Naadimuthu, G.; Bellman, R. E.; Wang K. M.; Lee, E. S. (1984), “Differential quadrature and partial differential equations: some numerical results”, J. Math. Anal. Appl. **98**, 220-235.

[119] Ni, Guoxi (1984), Ordinary Matrix Theory and Methods (in Chinese), Shanghai Sci. & Tech. Press.

[120] Noguchi, H.; Hisada, T. (1995), “Integrated FEM formulation for Total/Updated-Lagrangian method in geometrically nonlinear problems”, JSME Int. J. Series A. **38**, 23-29.

[121] Ortega, J. M.; Rheinboldt, W. C. (1970), Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York.

[122] Pandey, M. D.; Sherbourne A. N. (1991), “Buckling of anisotropic composite plates under stress gradient”, J. Eng. Mech. **117**, 260-275.

[123] Park, K. C. (1975), “An improved stiffly stable method for direct integration of non-linear structural dynamic equations”, ASME Journal of Applied Mechanics 464-470.

[124] Phillips, G. M.; Taylor, P. J. (1973), Theory and Applications of Numerical Analysis, Academic press Inc. Chapter 9.

[125] Philip, J. D.; Philip, R. (1975), Methods of Numerical Integration, Academic Press Inc.

[126] Quan, J. R.; Chang, C. T. (1989a), “New insights in solving distributed system equations by the quadrature methods - I”, Comput. Chem. Engrg., **13**, 779-788.

[127] Quan, J. R.; Chang, C. T. (1989b), “New insights in solving distributed system equations by the quadrature methods - II”, Comput. Chem. Engrg., **13**, 1017-1024.

[128] Roebuck, P. A.; Barnett, S. (1978), “A survey of Toeplitz and related matrices”, Int. J. Systems Sci., **8**, 921-934.

[129] Sherbourne, A. N.; Pandy, M. D. (1991), “Differential quadrature method in the buckling analysis of beams and composite plates”, Comput. Struct. **40**, 903-913.

[130] Shi, J. (1980), “Relationship between the elementary transformation and condition number”, J. Dalian Tech. Institute (in Chinese), **19**, 75-90.

[131] Shizgal, B.; Blackmore, R. (1984), “A discrete ordinate method of solution of linear boundary and eigenvalue problems”, J. Comput. Phys. **55**, 313-327.

[132] Shu, C.; Richards, B. E. (1992a), “Application of generalized differential quadrature to solve two-dimensional incompressible Navier-Stokes equations”, Int. J. Numer. Methods Fluids, **15**, 791-798.

[133] Shu, C.; Richards, B. E. (1992b), “Parallel simulation of incompressible viscous flows by generalized differential quadrature”, Computing Systems in Engineering, **3**, 271-281.

[134] Shu, C.; Khoo, B. C.; Yeo, K. S.; Chew, Y. T. (1994a), “Application of GDQ scheme to simulate natural convection in a square cavity”, Int. Commun. Heat Mass Transfer **21**, 809-817.

[135] Shu. C.; Khoo, B. C.; Yeo, K. S. (1994b), “Numerical solutions of incompresssible Navier-Stokes equations by generalized differential quadrature”, Finite Elements in Anal. and Des. 18, 83-97.

[136] Singh, G.; Sharma, A. K.; Rao, G. V. (1990), “Large-amplitude free vibrations of beams-a discussion on various formulations and assumptions”, J. Sound & Vibr., **142**, 77-85.

[137] Song, Y. D.; Middleton, R. H. (1992), “Dealing with time-varying parameter problem of robot manipulators performing path tracking tasks”, IEEE Trans. Auto. Contr. **37**, 1597-1601.

[138] Striz, A. G.; Jang, S. K.; Bert, C. W. (1988), “Nonlinear bending analysis of thin circular plates by differential quadrature”, Thin-Walled Struct. **6**, 51-62.

[139] Striz, A. G.; Chen, W.; Bert, C. W. (1994a), "Application of the differential quadrature method to the driven cavity problem", Int. J. Non-Linear Mech. **29**, 665-670.

[140] Striz, A. G.; Chen, Weilong; Bert, C. W. (1994b), "Static analysis of structures by the quadrature element method (QEM)", Int. J. Solids Structures **31**, 2807-2818.

[141] Striz, A. G.; Wang, X.; Bert, C. W. (1995), "Harmonic Differential Method and Applications to Structural Components", Acta Mechanica, **111**, 85-94.

[142] Szabo, B. A.; Lee, G. C. (1969), "Derivation of stiffness matrices for problems on plane elasticity by Galerkin methods", Int. J. Numer. Methods Eng. **1**, No. 3.

[143] Timoshenko, S.; Woinowsky-Krieger, S. (1959), Theory of plates and shells, 2nd Ed. McGraw-Hill, New York.

[144] Villadsen, J. (1970), Selected Approximation Methods for Chemical Engineering Problems: Reproset, Copenhagen.

[145] Wang, K. M. (1982), "Solving the model of isothermal reactors with axial mixing by the differential quadrature method", Int. J. Numer. Methods Eng. **18**, 111-118.

[146] Wang, T. M. (1969), "Nonlinear-linear bending of beams with uniformly distributed loads", Int. J. Non-linear Mech. **4**, 389-395.

[147] Wang, X.; Bert, C. W. (1993a), "A new approach in applying differential quadrature to static and free vibrational analyses of beams and plates", J. Sound & Vibration, **162**, 566-572.

[148] Wang, X.; Bert, C. W.; Striz, A. G. (1993b), "Differential quadrature analysis of deflection, buckling, and free vibration of beams and rectangular plates", Computers & Structures **48**, 473-479.

[149] Wang, X.; Bert, C. W.; Striz, A. G. (1993c), "Free vibration analysis of annular plates by the DQ method", J. Sound & Vibr. **164**, 173-175.

[150] Wang, X. ; Striz, A. G. ; Bert, C. W. (1994a), "Bucking and vibration analysis of skew plates by the DQ method", AIAA J. **37,** 886-889.

[151] Wang, X. (1994b), "The differential quadrature solution of nonlinear deflection of circular plate, (in Chinese)", J. Nanjing Univ. of Aeronautics & Astronautics, **26**, 172-175.

[152] Wang, X. (1994c), Private Communication.

[153] Wang, X. (1995), "Differential quadrature in the analysis of structural components (in Chinese)", Advances in Mechanics, **25**, 232-240.

[154] Wang, X.; He, B.; and Chen Wen (1996), "On the reducibility of centrosymmetric and skew centrosymmetric linear algebraic equations", J. Nanjing Univ. of Aeronautics & Astronautics, **28**, 599-607.

[155] Wang, Y.; Wang, Z. (1994), "Periodic response of piecewise-linear oscillators using trigonometric collocation", Journal of Sound & Vibration, **177**, 573-576.

[156] Wilkinson, J. H. (1965), The Algebraic Eigenvalue Problems, Oxford University Press.

[157] Woinowsky-Krieger, S. (1950), "The effect of an axial force on the vibration of hinged bars", Journal of Applied Mechanics **17**, 35-36.

[158] Wright, K. (1964), "Chebyshev collocation methods for ordinary differential equations", Comput. Jl, **6**, 358-365.

[159] Xu, Z. (1987), Method of Weighted Residuals in Solid Mechanics (in Chinese), Tongji Univ. Press, Shanghai.

[160] Yang, T. Y. (1972), "Finite displacement plate flexure by the use of matrix incremental approach", Int. J. Numer. Meths. Engrg. **4**, 415-432.

[161] Yin, Y. (1987), Introduction to Nonlinear Finite Element in Solid Mechanics (in Chinese), Peking Univ. Press, Beijing.

[162] Young, D. (1950), "Vibration of Rectangular Plates by Ritz Method", J. Appl. Mech. **17**, 448-453.

[163] Yuan, Z.; Fei J.; Liu D. (1987), Numerical methods for stiff initial value problems of ordinary differential equations (in Chinese), Sci. Press.

[164] Zhong, Wanxie; Lin, Jiahao; Qiu, Chunhang (1992), “Computational structural mechanics and optimal control - the simulation of substructural chain theory to linear quadrature optimal control problems”, Int. J. Numer. Methods Eng. **33**, 197-211.

[165] Zhong, Wanxie (1995), “Precise computation for transient analysis (in Chinese)”, Computational structural mechanics and applications, **12**, 1-6.

[166] Zhu, G.; Wang, H. (1989), “Quasi-conforming penalty FEM for large deflection of composite laminated plate (in Chinese)”, ACTA MATERIAE COMPOSITE SINICA, **6**, 39-47.

[167] Zwillinger, D. (1992), Handbook of differential equation, Academic Press, San Diego.

APPENDIX A.
On Algebraic and Analysis Properties of Hadamard product

There exists some significant algebraic and analysis properties in the Hadamard product, power and function, which are found similar to those in elementary algebra and scalar function. Section 2.2 has stated some properties of the Hadamard product. Horn's paper (1990) gave an comprehensive review on the Hadamard product and its applications. But we think that a systematic discussion from computing and analysis viewpoint may prove beneficial to further apply the Hadamard product to a wide range of nonlinear analysis. Horn (1990) discussed the properties of the Hadamard product involving norm inequality in great detail. Thus, we do not repeat the work of this aspect. The following brief discussions center on the algebraic and analysis properties of the Hadamard product, and let A, B, C, D, E, F, $X \in C^{n\times m}$, and $(\ )^{\circ(-1)}$ is denoted by $\frac{1}{(\ )}$.

Theorem 1. Multiplication and factorization

$$1\ (X + A) \circ (X + B) = X^{\circ 2} + (A + B) \circ X + A \circ B$$

$$2\ (A \pm B)^{\circ 2} = A^{\circ 2} \pm 2A \circ B + B^{\circ 2}$$

Theorem 2. Fraction

If the following Hadamard power and inverse exist, we have

$$1\ \frac{1}{(X - A) \circ (X - B)} = \frac{1}{(A - B)} \circ \left(\frac{1}{(X - A)} - \frac{1}{(X - B)}\right)$$

$$2\ \frac{P^{\circ}(X)}{(X - A) \circ (X - B) \circ (X - C)} = \frac{A_1}{(X - A)} + \frac{A_2}{(X - B)} + \frac{A_3}{(X - C)}$$

in which A_1, A_2 and A_3 are the desired constant matrices, P(x) is a polynomial.

Theorem 3. Radical expression

If n and p are scalar, $A^{\circ\frac{1}{n}}$ and $B^{\circ\frac{1}{n}}$ exist, we have

$$1\ \left(A^{\circ mp}\right)^{\circ\frac{1}{np}} = A^{\circ\frac{m}{n}}$$

$$2\ (A \circ B)^{\circ\frac{1}{n}} = A^{\circ\frac{1}{n}} \circ B^{\circ\frac{1}{n}}$$

Theorem 4. Rule of proportion

If $\frac{A}{B} = \frac{C}{D}$ and the related Hadamard inverses exist, we have

$$1.\ A \circ D = B \circ C$$

$$2.\ \frac{A \pm B}{B} = \frac{C \pm D}{D}$$

$$3.\ \frac{A + B}{A - B} = \frac{C + D}{C - D}$$

Theorem 5. Inequality

If A-B is positive semidefinite, we define A≥B; If A-B is positive definite. A ≻B. A≻0 means that A is positive definite, A≥0 denotes that A is positive semidefinite. In the following if A≻0, B≻0, A≻B, and all involved Hadamard inverse and power exist, we have

1> A°B≻0

2> A°C≻ B°C (C≻0) and A°C∠B°C (C≺0)

3> $\frac{A + B}{2} \geq (A \circ B)^{\frac{1}{2}}$

4> $\frac{A_1 + A_2 + \cdots + A_n}{n} \geq (A_1 \circ A_2 \circ \cdots \circ A_n)^{\frac{1}{n}}$ if A_i>0.

5> $\left(\frac{A_1 + A_2 + \cdots + A_n}{n}\right)^{\circ r} \prec \frac{A_1^{\circ r} + A_2^{\circ r} + \cdots + A_n^{\circ r}}{n}$, if r>1, A_i>0.

6> $\left(\frac{A_1 + A_2 + \cdots + A_n}{n}\right)^{\circ r} \succ \frac{A_1^{\circ r} + A_2^{\circ r} + \cdots + A_n^{\circ r}}{n}$, if r∠1, A_i>0.

7> If A≥0 and B≥0, then A°B≥0.

8> $A^{\circ 2}$≻0 irrespective of whether A≻0 or A∠0.

Formulas (1) and (7) are famous Schur inequality theorem (Ni, 1984; Horn, 1990) . The proofs for other inequalities are straightforward by using formulas (1) and (7) and the following Lemma I (Horn and Johnson, 1985):

Lemma I. If matrix A and B are positive definite or positive semidefinite, A+B is positive definite or positive semidefinite.

Theorem 6. Hadamard function

1. $sin^{\circ}(A \pm B) = sin^{\circ} A \circ cos^{\circ} B \pm cos^{\circ} A \circ sin^{\circ} B$
2. $cos^{\circ}(A \pm B) = cos^{\circ} A \circ cos^{\circ} B \mp sin^{\circ} A \circ sin^{\circ} B$
3. $e^{\circ(A+B)} = e^{\circ A} \circ e^{\circ B}$
4. $(e^{\circ A})^{\circ m} = e^{\circ mA}$

From the above formulas, it can be found that the Hadamard functions have the same properties as the scalar function.

Theorem 7. Differentiation and derivation

The differentiation properties of the Hadamard product on matrix-valued function of a variable vector or matrix are equivalent to those on the Jacobian derivative matrix, and thus can be found in section 2.3 involving the operation rules of the SJT product, namely, equations (2.3-4a, b, c, d, e, f). We herein discuss the properties of the Hadamard product involving matrix-valued function of a scalar variable. Such matrix-valued function is often encountered in perturbation theory (Horn, 1990) and the initial-boundary problems including time-derivative. The differentiation or integration of such function is the matrix obtained by entry-wise differentiation or integration.

Lemma II. If scalar function F(x) and matrix-valued functions $A=[a_{ij}(t)]$ of a real variable t are continuous, differentiable and integrable, the Hadamard function $F^{\circ}(A)$ is continuous, differentiable and integrable.

In the following $\frac{d}{dt}$ is also represented by $(\ )'$,

1. $\frac{dC \circ A(t)}{dt} = C \circ \frac{dA(t)}{dt}$, where C is a constant matrix.
2. $\frac{dA(t) \circ B(t)}{dt} = \frac{dA(t)}{dt} \circ B(t) + A(t) \circ \frac{dB(t)}{dt}$
3. $\frac{df^{\circ}(A(t))}{dt} = \frac{df^{\circ}(A)}{dA}\frac{dA(t)}{dt}$

For example,

$$\left(A^{\circ n}\right)' = nA^{\circ(n-1)} \circ A', \qquad \left(In^{\circ} A\right)' = A^{\circ(-1)} \circ A', \qquad \left(e^{\circ A}\right)' = e^{\circ A} \circ A',$$

$$\left(\sin^{\circ} A\right)' = \left(\cos^{\circ} A\right) \circ A', \qquad \left(\cos^{\circ} A\right)' = \left(-\sin^{\circ} A\right) \circ A',$$

$$\left(sh^{\circ} A\right)' = \left(ch^{\circ} A\right) \circ A', \qquad \left(ch^{\circ} A\right)' = \left(sh^{\circ} A\right) \circ A'$$

where A=A(t), In( ) denotes natural logarithm function, sh( ) and ch( ) represent hyperbolic sine and cosine functions, respectively.

4. $\int_{t_0}^{t_1} C \circ A(t)dt = C \circ \int_{t_0}^{t_1} A(t)dt$, where C is a constant matrix.

APPENDIX B.

Circulant Matrix Structures in the Weighting Coefficient Matrices of the Quadrature Method Based on the Fourier-type Trigonometric Interpolation

In chapter 6 we have proven the centrosymmetric structures of weighting coefficient matrix of the DQ and HDQ methods under conditions of symmetric grid spacing. In this appendix we will show that the weighting coefficient matrices possess circulant structures in the quadrature method using the Fourier interpolation principle. For simplicity, the quadrature method is denoted as the FDQ method.

The Fourier trigonometric basic functions are equal to those in equation (1.2-6) for the HDQ method. The only difference between the HDQ and so-called FDQ methods is the coordinates of sampling points when the equally spaced grid points are used. The HDQ method uses the spacing points as sampling points, while the FDQ method employs middle inner points between the respective two spacing points as its sampling points, which is similar to the choice of grid points in the discrete Fourier transformation (DFT) approach. To clearly illustrate the difference, the grid points for the HDQ and FDQ methods are shown in Figs. 1 and 2, respectively.

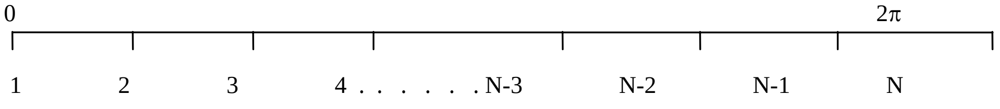

Fig. 1. Equally spaced grid points for the HDQ method

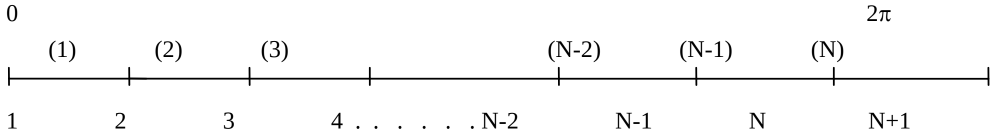

Fig. 2. Equally spaced grid points for the FDQ method

The numbers with parentheses in Figure 2 indicate the sequence of sampling points in the FDQ method. Obviously, the grid points in the Fourier interpolation principle do not include the boundary points, but this does not matter. A simple algebraic transformation can overcome the difficulty as we have done for the zeros of orthogonal polynomials in section 3.3. Applying equation (6.2-13) and some properties of trigonometric functions, we obtain the following direct computing formulas for computing the FDQ weighting coefficients of the 1st order derivative in the normalized domain (0≤x≤1), namely,

$$A_{ij} = \frac{(-1)^{i-j}}{\sin\frac{(i-j)}{N}\pi}\frac{N-1}{N}\pi, \ \mathrm{i{\neq}j} \qquad \text{(B1)}$$

and

$$A_{ii} = 0, \qquad \text{(B2)}$$

where N is the number of grid points. Considering the completeness requirements in the choice of the test functions, N is in general odd integer. Here the coordinates of grid points are

$$x_i = \frac{i-1}{N-1}. \qquad \text{(B3)}$$

It is stressed that the FDQ method usually employs the equally spaced grid points. According to equations (B1) and (B2), the weighting coefficient matrix [A] is a circulant one. It is known that the ordinary product of two circulant matrices produces a circulant one. The inverse of a circulant is also circulant one. In section 1.2 we have shown that the weighting coefficients of the DQ-type methods can be obtained by using successive multiplications of the [A] matrix by itself, namely, equation (1.2-7). Therefore, it is straightforward that the FDQ weighting coefficient matrices are in general circulant matrix for the various order derivatives. For example, considering five equally spaced grid points, we have

$$[A_{ij}]=\begin{bmatrix} 0.0 & 4.28 & -2.64 & 2.64 & -4.28 \\ -4.28 & 0.0 & 4.28 & -2.64 & 2.64 \\ 2.64 & -4.28 & 0.0 & 4.28 & -2.64 \\ -2.64 & 2.64 & -4.28 & 0.0 & 4.28 \\ 4.28 & -2.64 & 2.64 & -4.28 & 0.0 \end{bmatrix} \qquad \text{(B4)}$$

and

$$[B_{ij}]=\begin{bmatrix} -50.5 & 29.6 & -4.32 & -4.32 & 29.6 \\ 29.6 & -50.5 & 29.6 & -4.32 & -4.32 \\ -4.32 & 29.6 & -50.5 & 29.6 & -4.32 \\ -4.32 & -4.32 & 29.6 & -50.5 & 29.6 \\ 29.6 & -4.32 & -4.32 & 29.6 & -50.5 \end{bmatrix}, \qquad \text{(B5)}$$

where [B_{ij}] denotes the weighting coefficient matrix for the 2nd order derivative. The number of independent elements in the weighting coefficient matrices of N order are N. In fact, there is only (N-1)/2 independent elements for the odd order derivatives and (N+1)/2 ones for the even order derivatives. The weighting coefficient matrices in the FDQ method are also real, symmetric or skew symmetric finite Toeliptz matrices. The circulant structures of the FDQ weighting coefficient matrices are closely related to the periodic behavior of trigonometric functions. The circulant matrix as well as symmetric finite Toeliptz matrices has many interesting properties (Cheng, 1989; Roebuck and Barnett, 1978). Some very efficient algorithms have been well developed for solving such kind of matrices (Morgera, 1982; Cantoni and Butler, 1976; Cheng, 1989; Roebuck and Barnett, 1978). The further work in this direction may be very beneficial to reduce computation effort and storage requirements in the FDQ method.

We apply the FDQ method to analysis the free vibration of a simply supported beam. The corresponding FDQ formulation and solution procedures are the same as those in the example 1 of section 6.3. The FDQ solutions[1] using thirteen grid points are tabulated in table 1.

Table 1. FDQ solutions of free vibration of a simply supported beam ($\varpi^2=\rho A_0 L^4 \omega^2/\pi^2 EI$)

n	1	2	3	4	5	6	7	8	9	10	11
exact	1	4	9	16	25	36	49	64	81	100	121
FDQ	1.03	4.20	9.27	16.8	25.7	37.95	50.3	67.8	82.7	106.8	122.3

n in table 1 means the modal order. It is noted that the FDQ solutions for the odd order modals are obviously better than those for the even odd order modals. Moreover, as the modal increases, the accuracies improves. This phenomena is abnormal and does not occur in the HDQ and DQ solution of the similar problems. In this case, the HDQ method gives the exact solution, but the HDQ method yields the significant real solutions only in the first several order modals. Therefore, it is concluded that the FDQ method may be advantageous for the cases in which the high order modals are desired. In addition, the symmetric Toeliptz matrix structures in the FDQ weighting coefficient matrices may be more valuable for reducing computational and storage requirements than the centrosymmetric structures in the DQ and HDQ methods (Cantoni and Butler, 1976; Cheng, 1989).

In addition, it is worth pointing out that the FDQ method is in fact equivalent to the discrete Fourier transformation approach in the solution of differential equations. The only difference between them is that the desired values in the FDQ method are the unknown function values rather than the spectral coefficients as in the DFT method. Direct computing the unknown function values has somewhat advantage in some situations. Especially if the desired spectral coefficients for the considered problems have no practical physical significance, the DFT technique will require an inverse Fourier transformation and suffer from the difficulty in the choice of starting guess in the iteration solution of nonlinear problems. In addition, the FDQ weighting coefficients need to be computed only once and are independent of any special problems. Therefore, these weighting coefficients can be used repeatedly for a variety of problems. The FDQ method may be a potential alternative to the DFT technique in the solution

[1] The FDQ solutions for this case were provided by professor Xinwei Wang (1994c).

of some differential systems in engineering such as electromagnetism, electronic analysis, and computing microwave, etc. The subject is currently in active study.

LIST OF PUBLICATIONS

[1] Wen Chen, Yongxi Yu and X. Wang, “Reducing the Computational Requirement of the Differential Quadrature Method (in English)”, Numerical Methods for Partial Differential Equations, Vol. 12, 565-577, 1996.

[2] Wen Chen, X. Wang and Tingxiu Zhong, “The Structure of Weighting Coefficient Matrices of Harmonic Differential Quadrature and Its Applications (in English)”, Communications in Numerical Methods in Engineering, Vol. 12, No. 8, 455-560, 1996.

[3] Wen Chen and Tingxiu Zhong, “The study on the nonlinear computations of the DQ and DC methods (in English)”, Numerical Methods for Partial Differential Equations, (in press) 1996.

[4] Wen Chen, Tingxiu Zhong and W. He, “A Note on the DQ Analysis of Anisotropic Plates (in English)”, J. Sound & Vibration (accepted), 1996.

[5] Wen Chen, Tingxiu Zhong and Jianwu Zhang, “Hadamard Product and Numerical Nonlinear Computations−1: Formulations (in English)”, Inter, J. Numer. Methods Eng. (to appear), 1996.

[6] Wen Chen, Tingxiu Zhong and Jianwu Zhang, “Hadamard Product and Numerical Nonlinear Computations−2: Further Analysis and applications (in English)”, Inter, J. Numer. Methods Eng. (to appear), 1996.

[7] Wen Chen, Tingxiu Zhong and S. Liang, “Structural Dynamic Analysis Using Differential Quadrature Method (in English)”, J. Sound & Vibration (to appear) 1996.

[8] Wen Chen, Tingxiu Zhong and W. He, “The DQ Solution of Geometrically Nonlinear Bending of Orthotropic Rectangular Plates by Using Hadamard and SJT Product (in English)”, Computers & Structures (to appear) 1996.

[9] Wen Chen and Tingxiu Zhong, “Minimized Computational Effort in Applying the DQ Method to the Poisson and Convection-Diffusion Equations (in English)”, J. Comput. Phys. (to appear) 1996.

[10] Wen Chen, S. Liang and Tingxiu Zhong, “On the DQ Analysis of Geometrically Nonlinear Vibration of Immovably Simply Supported Beams (in English)”, J. Sound & Vibration (to appear) 1996.

[11] Wen Chen and Tingxiu Zhong, “Comments on application of the differential quadrature method to the driven cavity problem by Striz and Chen”, Inter. J. Numer. Nonlinear Mech. (to appear), 1996.

[12] 陈文， 钟廷修，郁永熙， “特殊矩阵乘积在非线性数值计算中的应用”，中国科学通报（已投）1996.

[13] 王鑫伟，何柏庆，陈文，“中心或反中心对称线性方程组的缩减算法”，南京航空航天大学学报, Vol. 28, No. 599-607, 1996.

[14] 梁枢平，李宏峰，陈文，“DQ法解圆拱在静水压力下的稳定问题”Vol. 25, No. 1, 81-82, 1997.

1.	Guanghua Scholarship	1994-1995
2.	Guanghua Scholarship	1995-1996
3.	Siemens Prize	1995-1996
4.	Motorola Priz	1996-1997