Copyright © 2011 Tech Science Press CMES, vol.71, no.2, pp.157-178, 2011

A New Insight into the Differential Quadrature Method in Solving 2-D Elliptic PDEs

Ying-Hsiu Shen[1] **and Chein-Shan Liu**[1,2]

Abstract: When the local differential quadrature (LDQ) has been successfully applied to solve two-dimensional problems, the global method of DQ still has a problem by requiring to solve the inversions of ill-posed matrices. Previously, when one uses $(n-1)$th order polynomial test functions to determine the weighting coefficients with n grid points, the resultant $n \times n$ Vandermonde matrix is highly ill-conditioned and its inversion is hard to solve. Now we use $(m-1)$th order polynomial test functions by n grid points that the size of Vandermonde matrix is $m \times n$, of which m is much less than n. We find that the $(m-1)$th order polynomial test functions are accurate enough to express the solutions, and the novel method significantly improves the ill-condition of algebraic equations. Such a new DQ as being combined with FTIM (Fictitious Time Integration Method) can solve 2-D elliptic type PDEs successfully. There are some examples tested in this paper and the numerical errors are found to be very small.

Keywords: Differential quadrature (DQ), Vandermonde matrix, Fictitious time integration method (FTIM), Dirichlet boundary conditions, Elliptic Partial differential equations

1 Introduction

Generalized differential quadrature (GDQ) was invented by Bellman and Casti (1971), and it has been an useful numerical technique to treat the differential terms appeared in differential equation. It follows the concept of classical integral quadrature and the idea is to use the weighted sum of all function values at all selected grid points to express the derivatives at a point we interest. It is also called a global method as shown by Shu (2000). As being compared with local method of numerical techniques, like as finite element method and lower-order finite difference

[1] Department of Civil Engineering, National Taiwan University, Taipei, Taiwan
[2] Corresponding author. E-mail: liucs@ntu.edu.tw

method, DQ has high accuracy by only needing a small number of grid points distributed inside the problem domain.

The key procedure in DQ is to properly determine the weighting coefficients. There are two ways to carry this out. The first way is the polynomial test functions, which were originally introduced by Bellman, Kashef and Casti (1972). This system has a unique solution because its coefficent matrix is of Vandermonde form, and the determinant of Vandermonde matrix is a non-zero value. Unfortunately, when the number of discrete grid points is large, the matrix is highly ill-conditioned and its inverse is difficult to find. The second method was proposed by Quan and Chang (1989a, 1989b) that using the Lagrange interpolation to express the explicit formulas for the weighting coefficients. But this method was also not useful when the number of grid points is large. Because of the limited applications of DQ, the localized DQ (LDQ) was proposed by Zong and Lam (2002). It is used by applying the DQ approximation to a small number of grid points inside a neighborhood of the point we interest rather than to all the grid points in the whole domain. The derivative at each selected point is the weighted sum of the function values of some points inside each neighborhood instead of all of points.

In this paper, we propose a new insight into the differential quadrature. For n grid points, we use only $(m-1)$th order polynomial test functions instead of $(n-1)$th order polynomial test functions, which were originally introduced by Bellman, Kashef and Casti (1972), of which m is much less than n. We use the improved polynomial test functions and obtain the weighting coefficients by using the conjugate gradient method (CGM) [Liu, Hong and Atluri (2010)] to solve the underdeterminate linear algerbraic equations. We find that the $(m-1)$th order polynomial test functions are accurate enough to express the solutions and the novel method significantly improves the ill-condition of algebraic equations.

Finally the new DQ is combined with the Fictitious Time Integration Method (FTIM) proposed by Liu and Atluri (2008) to solving some problems. Five examples are given to show its improved efficacy in one-dimensional problems, Cauchy problem and elliptic type partial differential equations (PDEs). The numerical errors are found to be very small as compared to the exact solutions.

2 A new insight in global DQ

2.1 An improved polynomial test functions in DQ

According to the concept of DQ, the first order derivative of a differentiable function $f(x)$ with respect to x at a point x_i, is approximately expressed as

$$f'(x_i) = \sum_{j=1}^{n} a_{ij} f(x_j), \tag{1}$$

where a_{ij} is the weighting coefficient contributed from the j-th grid point to the first order derivative at the i-th grid point. Similarly, in DQ the second order derivative at the i-th grid point is given by

$$f''(x_i) = \sum_{j=1}^{n} b_{ij} f(x_j), \tag{2}$$

where $b_{ij} = \sum_{k=1}^{n} a_{ik} a_{kj}$ is the weighting coefficient of the second order derivative. In order to determine the weighting coefficients, the test functions were proposed by Bellman, Kashef and Casti (1972). They used the polynomials as test functions with orders from zero to $n-1$ when the number of grid points is n, that is,

$$g(x_i) = x_i^{k-1},\ i = 1, \ldots, n,\ \ k = 1, \ldots, n. \tag{3}$$

Then inserting the test functions $g(x) = x^{k-1}$ for $f(x)$ into Eq. (1) leads to a set of linear algebraic equations:

$$\begin{bmatrix} 1 & 1 & \ldots & 1 & 1 \\ x_1 & x_2 & \ldots & x_{n-1} & x_n \\ x_1^2 & x_2^2 & \ldots & x_{n-1}^2 & x_n^2 \\ \vdots & \vdots & \ldots & \vdots & \vdots \\ x_1^{n-2} & x_2^{n-2} & \ldots & x_{n-1}^{n-2} & x_n^{n-2} \\ x_1^{n-1} & x_2^{n-1} & \ldots & x_{n-1}^{n-1} & x_n^{n-1} \end{bmatrix} \begin{bmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{ik} \\ \vdots \\ a_{in} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ k x_i^{k-1} \\ \vdots \\ (n-1) x_i^{n-2} \end{bmatrix}. \tag{4}$$

We can obtain the weighting coefficients a_{ij} from solving the above equations, but this linear system has the Vandermonde form. The ill-posedness of Vandermonde system is higher when the number of grid points is larger. Usually, the solutions were not accurate when the number of grid points was larger than 13. In order to overcome the ill-posedness, a local DQ was thus proposed. However, local DQ was only appropriate in small number of grid points.

Because of the limited applications in DQ, we propose a new insight into the differential quadrature. As being a difference from the original $(n-1)$th order polynomial test functions, we use $(m-1)$th order polynomial test functions by n grid points, of which m is less than n. Then the improved test functions can be expressed as

$$g(x_i) = x_i^{k-1},\ i = 1,\dots,n,\ \ k = 1,\dots,m. \tag{5}$$

Correspondingly, we can obtain an under-determinate linear algebraic equations system:

$$\begin{bmatrix} 1 & 1 & \dots & 1 & 1 \\ x_1 & x_2 & \dots & x_{n-1} & x_n \\ x_1^2 & x_2^2 & \dots & x_{n-1}^2 & x_n^2 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ x_1^{m-2} & x_2^{m-2} & \dots & x_{n-1}^{m-2} & x_n^{m-2} \\ x_1^{m-1} & x_2^{m-1} & \dots & x_{n-1}^{m-1} & x_n^{m-1} \end{bmatrix} \begin{bmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{ik} \\ \vdots \\ a_{in} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ kx_i^{k-1} \\ \vdots \\ (m-1)x_i^{m-2} \end{bmatrix}. \tag{6}$$

For solving the under-determinate linear algebraic equations, we use the least-square method and conjugate gradient method (CGM) [Liu, Hong and Alturi (2010)] to obtain the weighting coefficients a_{ij}, where the least-square method is

$$\min \|\mathbf{Ax}-\mathbf{b}\| \Rightarrow \mathbf{A}^{\mathrm{T}}\mathbf{Ax} = \mathbf{A}^{\mathrm{T}}\mathbf{b}. \tag{7}$$

The flow chart of CGM is given in Fig. 1 for the linear algebraic equations $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is a positive matrix, $\mathbf{r}$ is the vectorial residual error, and α, η and $\mathbf{p}$ are variables. The convergence criterion is fixed to be $\varepsilon = 10^{-12}$.

2.2 Determination of the order of the improved polynomial test functions

In order to determine a suitable order of the improved polynomial test functions, we use three common functions e^x, $\sin x$ and $\cos x$ to decide m, which is the order of the improved test functions. In the linear ODEs and PDEs, the solutions sometimes can be expressed by a combination of these three functions. Then we compare the exact solutions of their first order derivatives with the approximate solutions obtained from differential quadrature with the improved test functions from $m = 2$ to $m = n$. The exact solutions $f_e'(x)$ and the approximate solutions $f_a'(x)$ are expressed below only for a demonstrative case of $f(x) = \sin x$,

$$\begin{aligned} &f_e'(x_i) = \cos x_i,\ \ i = 1,\dots,n, \\ &f_a'(x_i) = a_{i1}\sin x_1 + \dots + a_{in}\sin x_n,\ \ i = 1,\dots,n. \end{aligned} \tag{8}$$

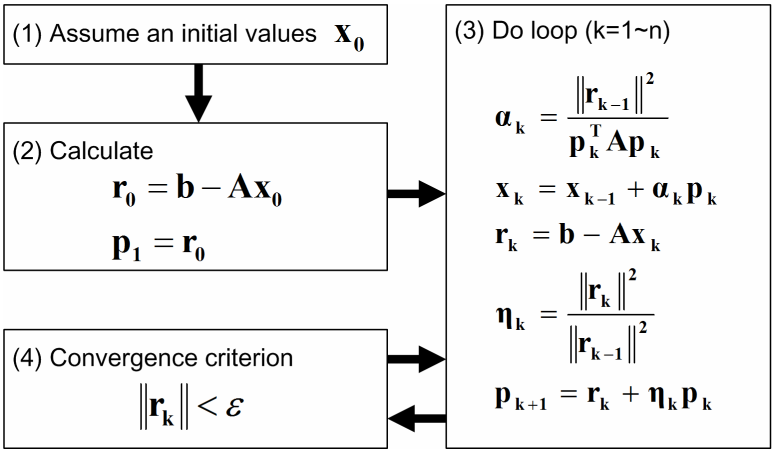


Figure 1: The flow chart of CGM.

We define the maximum error via

$$\varepsilon_c = \max_{i=1}^{n} \varepsilon_i = \max_{i=1}^{n} |f_a'(x_i) - f_e'(x_i)|. \tag{9}$$

First, we discretize the domain of $[a,b]$ by n grid points, and each grid point is read as

$$x_i = \frac{i-1}{n-1}(b-a) + a, \quad a \le x_i \le b. \tag{10}$$

Here, we fix $a = 0$ and $b = 1$. First, we set $n = 30$ and obtain the weighting coefficients by CGM. The errors ε_c from $m = 2$ to $m = n$ are, respectively, shown in Fig. 2 for the above three selected functions. The smallest errors of these three functions are 1.39×10^{-13}, 4.127×10^{-14} and 7.25×10^{-14} when $m = 12$. Obviously, the higher order test functions do not have higher accuracy. So we can decide $m = 12$ when we set 30 grid points in the range of the domain $0 \le x \le 1$. The highest order of the improved polynomial test functions is 11.

Then we increase the number of grid points to $n = 40$ to determine m, and we also fix the domain to be $0 \le x \le 1$. The errors ε_c from $m = 2$ to $m = n$ are shown in Fig. 3. The smallest errors of these three functions by 40 grid points are 1.992×10^{-10}, 3.555×10^{-10} and 8.465×10^{-10} when $m = 10$. So we decide to use $m = 10$ by 40 grid points. The highest order of the improved polynomial test functions is 9. Lastly, we change the number of grid points to $n = 50$ to determine m. The errors ε_c from $m = 2$ to $m = 50$ are, respectively, shown in Fig. 4 for the above

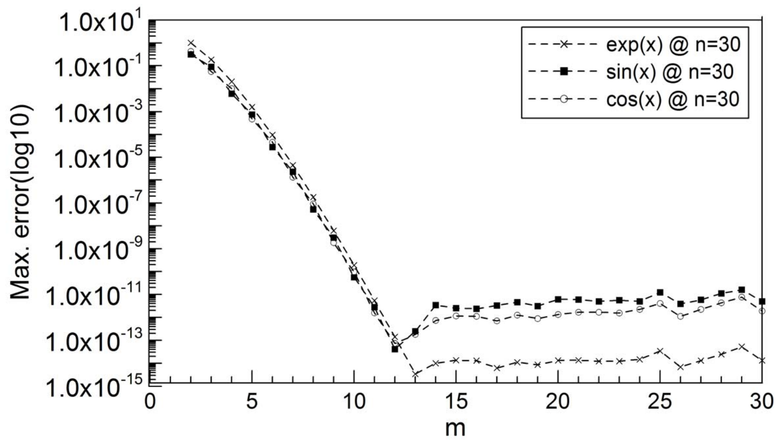


Figure 2: The errors of three functions from $m = 2$ to $m = 30$ (when $n = 30$) in $0 \leq x \leq 1$.

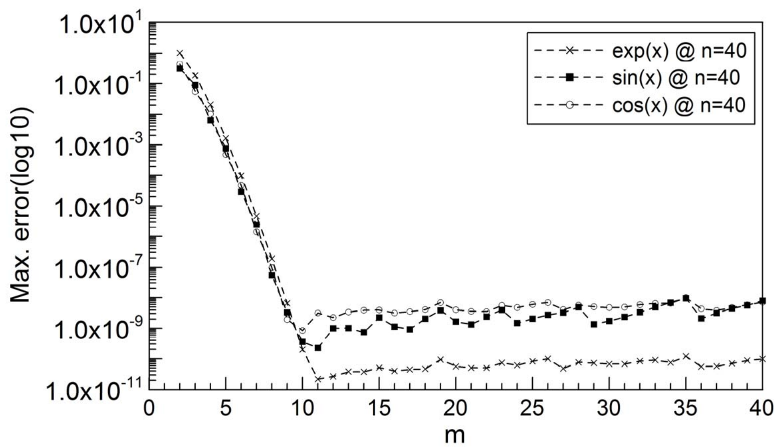


Figure 3: The errors of three functions from $m = 2$ to $m = 40$ (when $n = 40$) in $0 \leq x \leq 1$.

three selected functions. The smallest errors of these three functions by 50 grid points are 2.086×10^{-10}, 3.968×10^{-10} and 9.294×10^{-10} when $m = 10$. So we determine $m = 10$ when 50 grid points are used. The highest order of the improved polynomial test functions is 9. The results indicate that the order of the improved polynomial test functions is influenced by the number of grid points. More grid points may lose the accuracy, so $n = 30$ and $m = 12$ are the best choice.

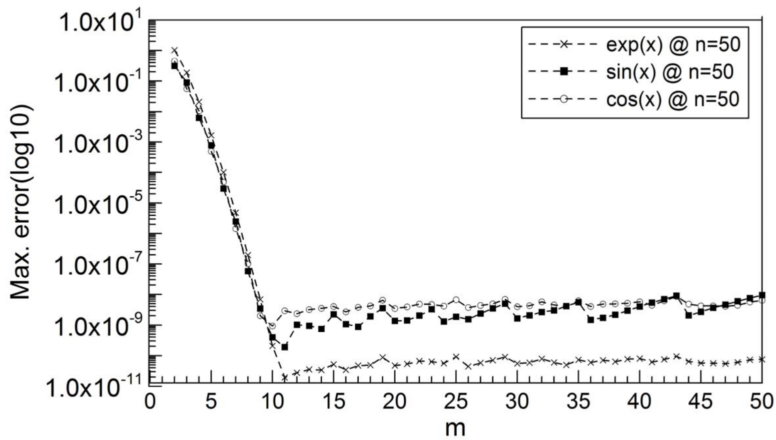


Figure 4: The errors of three functions from $m = 2$ to $m = 50$ (when $n = 50$) in $0 \leq x \leq 1$.

3 Fictitious time integration method

The Fictitious Time Integration Method (FTIM) was proposed by Liu and Atluri (2008). Its central idea is adding a fictitious time and transforming the linear or non-linear algebraic equations (NAEs) into a system of nonautonomous first order ODEs.

We consider a linear or nonlinear algebraic equations system:

$$F_i(x_1, \ldots, x_n) = 0, \quad i = 1, \ldots, n. \tag{11}$$

First, we introduce the following transformation:

$$y_i(t) = (1+t)x_i. \tag{12}$$

Here, t is a variable which is independent of x_i; hence,

$$\dot{y}_i = dy_i/dt = x_i, \quad i = 1, \ldots, n. \tag{13}$$

If $\nu \neq 0$, Eq. (11) is equivalent to

$$0 = -\nu F_i(x_1, \ldots, x_n), \tag{14}$$

where ν is a nonzero constant.

Adding Eq. (13) and Eq. (14) we can obtain

$$\dot{y}_i = x_i - \nu F_i(x_1, \ldots, x_n), \quad i = 1, \ldots, n. \tag{15}$$

By using Eq. (12) we can derive

$$\dot{y}_i = \frac{y_i}{1+t} - \nu F_i\left(\frac{y_1}{1+t},\ldots,\frac{y_n}{1+t}\right), \quad i = 1,\ldots,n. \tag{16}$$

Multiplying Eq. (16) by an integrating factor of $1/(1+t)$ we can obtain

$$\frac{\dot{y}_i}{1+t} = \frac{y_i}{(1+t)^2} - \frac{\nu}{1+t}F_i\left(\frac{y_1}{1+t},\ldots,\frac{y_n}{1+t}\right), \quad i = 1,\ldots,n. \tag{17}$$

Then, we have

$$\frac{d}{dt}\left(\frac{y_i}{1+t}\right) = -\frac{\nu}{1+t}F_i\left(\frac{y_1}{1+t},\ldots,\frac{y_n}{1+t}\right), \quad i = 1,\ldots,n. \tag{18}$$

By using $y_i/(1+t) = x_i$ again, we can obtain the first order derivative of x_i:

$$\dot{x}_i = -\frac{\nu}{1+t}F_i(x_1,\ldots,x_n), \quad i = 1,\ldots,n. \tag{19}$$

We can get the solutions by using the fictitious time integration method when the following iterations reach a convergence:

$$x_i^{k+1} = x_i^k - \frac{h\nu}{1+t_k}F_i(x_1^k,\ldots,x_n^k), \quad i = 1,\ldots,n, \tag{20}$$

where h is the time stepsize, the subscript i denotes the i-th variable, and the superscript k denotes the k-th time step.

Compared with Newton method, the FTIM is not influenced obviously by the initial values. It also does not need to calculate the Jacobian matrix and its inverse. Because of the advantages of FTIM, we can solve the algebraic equations easily and do not spend much time on calculations.

4 Examples

Here, we combine the new DQ with the FTIM in some examples and compare the numerical solutions with exact solutions. These examples are all of the Dirichlet boundary problems, including linear and non-linear 1D problems, Cauchy problem and non-linear elliptic type PDEs. First we use the conjugate gradient method to calculate the weighting coefficients by the concept of the new DQ, and the convergence criterion used in CGM is $\varepsilon = 10^{-12}$. Then we solve the resultant algebraic equations by FTIM. In these problems, we also use a scaling factor R which was proposed by Liu and Atluri (2009), and it can also improve the ill-conditioned algebraic problems effectively. The results of the new DQ and the scaling factor in DQ are enclosed for comparison.

4.1 Example 1

In this example we apply the new DQ and FTIM to solve the boundary value problem of a linear second order differential equation:

$$2x^2y''(x) - 7xy'(x) + 10y(x) = 3x. \tag{21}$$

Under the boundary conditions $y(1) = 1$ and $y(2) = 4$, its exact solution is

$$y(x) = x + \frac{x^2(1-\sqrt{x})}{2-2\sqrt{2}}. \tag{22}$$

We choose $n = 30$ and use the new DQ to decide the order of the test functions, that is $m = 12$. The scaling factor $R = 1.8$ and the parameters in FTIM are $h = 0.1$, $\nu = -0.05$ and $\varepsilon = 10^{-6}$. The initial guess is $y_i = y(1) + (i-1)[y(2) - y(1)]/(n-1)$, $i = 1, \ldots, n$. It converges within 2992 steps when using the new DQ and within 5321 steps when using a scaling factor $R = 1.8$ in DQ. The convergence of residual error is shown in Fig. 5. The numerical results are shown in Fig. 6(a) and the numerical errors are displayed in Fig. 6(b). The maximum errors are 6.084×10^{-4} in the new DQ and 5.878×10^{-4} in the scaling DQ with $R = 1.8$. The errors in the new DQ are as small as that obtained from the DQ by using a scaling factor.

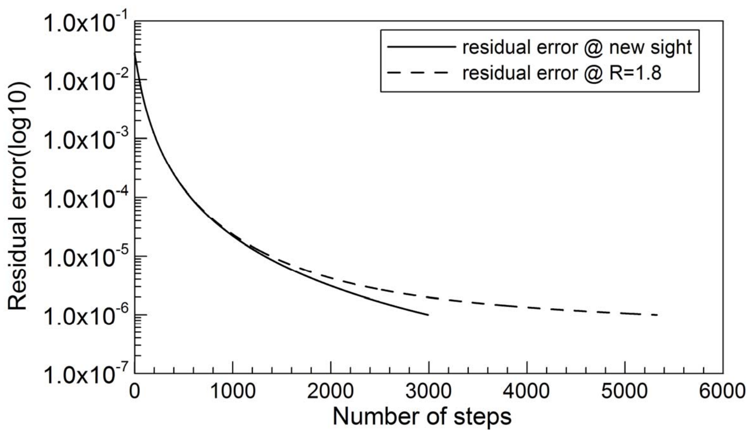


Figure 5: The convergence of residual error via number of steps with the new sight DQ and the scaling DQ with $R = 1.8$ for example 1.

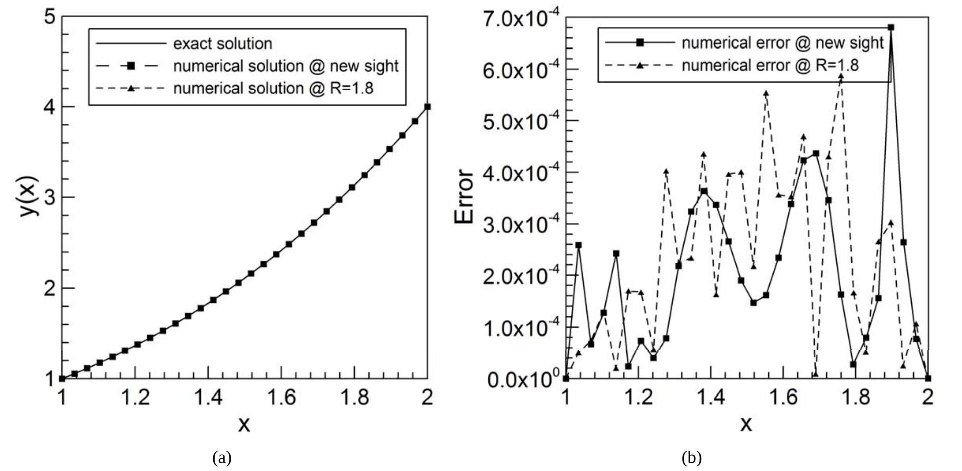


Figure 6: (a) Comparing numerical and exact solutions, and (b) numerical errors in the new DQ and the scaling DQ with $R = 1.8$ for example 1.

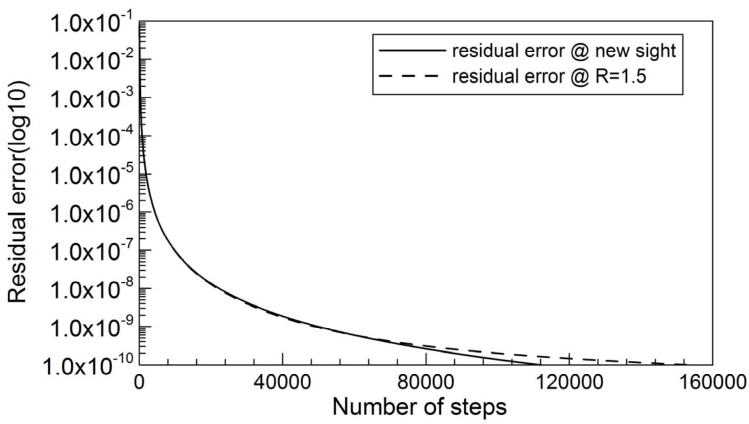
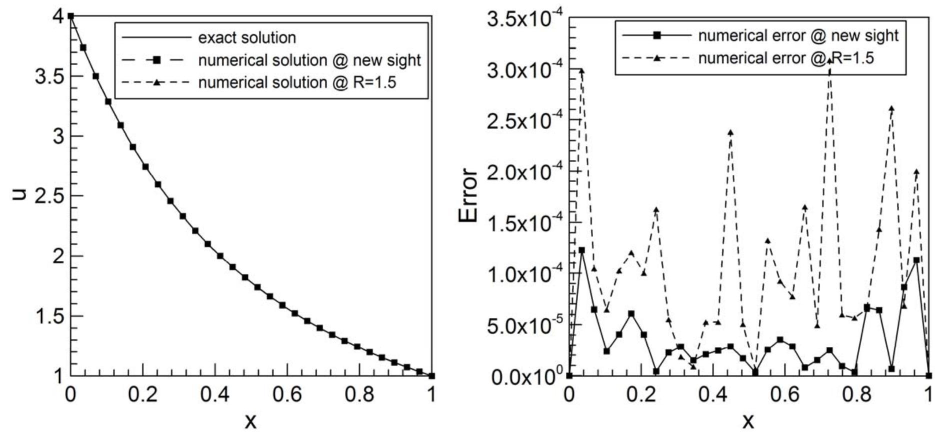


Figure 7: The convergence of residual error via number of steps with the new sight DQ and the scaling DQ with $R = 1.5$ when $n = 30$ for example 2.

4.2 Example 2

In this example we apply the new DQ and FTIM to solve the boundary value problem of a non-linear second order differential equation:

$$u''(x) = \frac{3}{2}u^2(x). \tag{23}$$



(a) (b)

Figure 8: (a) Comparing numerical and exact solutions, and (b) numerical errors in the new DQ and the scaling DQ with $R = 1.5$ when $n = 30$ for example 2.

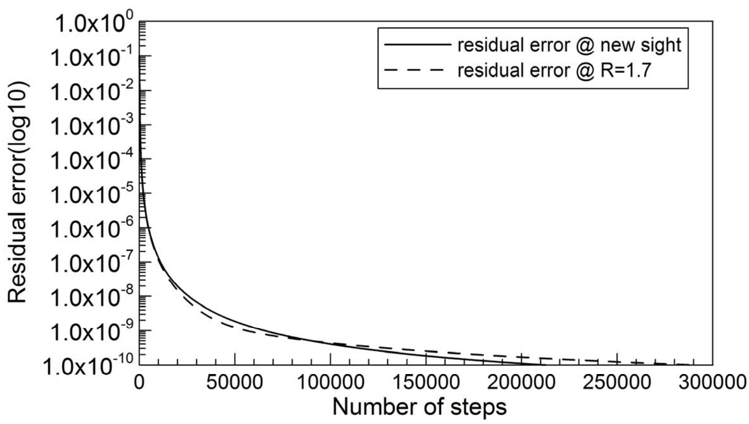
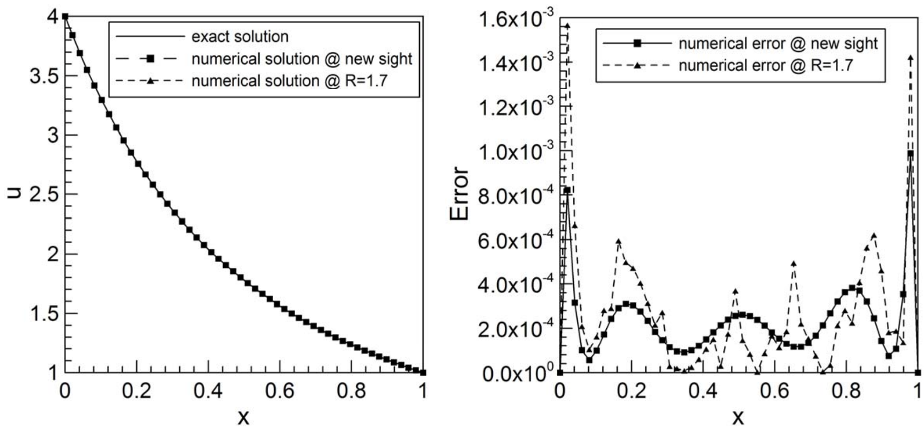


Figure 9: The convergence of residual error via number of steps with the new sight DQ and the scaling DQ with $R = 1.7$ when $n = 50$ for example 2.

Under the boundary conditions $u(0) = 4$ and $u(1) = 1$, the exact solution is

$$u(x) = \frac{4}{(1+x)^2}. \tag{24}$$

First we use $n = 30$ and $m = 12$ in the new DQ to obtain the weighting coefficients. The scaling factor $R = 1.5$ and the parameters in FTIM are $h = 0.01$, $\nu = -0.12$ and



(a) (b)

Figure 10: (a) Comparing numerical and exact solutions, and (b) numerical errors in the new DQ and the scaling DQ with $R = 1.7$ when $n = 50$ for example 2.

$\varepsilon = 10^{-10}$. The initial guess is $u_i = u(0)+(i-1)[u(1)-u(0)]/(n-1),\ i = 1,\dots,n$. It converges within 112180 steps when using the new DQ and within 154225 steps when using a scaling factor $R = 1.5$ in DQ. The convergence of residual error is shown in Fig. 7. The results and errors are shown in Figs. 8(a) and 8(b). The maximum errors are 1.23×10^{-4} in the new DQ and 3.084×10^{-4} in the scaling DQ with $R = 1.5$. Then we increase the number of grid points to $n = 50$ and $m = 10$ in the new DQ and the scaling factor is increased to $R = 1.7$. The parameters in FTIM remain the same. It converges within 221661 steps when using the new DQ and within 290829 steps when using a scaling factor $R = 1.7$ in DQ. The convergence of residual error is shown in Fig. 9. The results and errors are shown in Figs. 10(a) and 10(b). The maximum errors are 9.884×10^{-4} in the new DQ and 1.566×10^{-3} in the scaling DQ with $R = 1.7$. The result obtained by the new DQ is good and the error is very small.

4.3 Example 3

In this example we apply the new DQ and FTIM to solve the 2-D problem of the Cauchy problem, whose governing equation is

$$\frac{\partial u}{\partial x} + 3\frac{\partial u}{\partial y} = 2u,\ \ 0 \le x \le 1,\ 0 \le y \le 1. \tag{25}$$

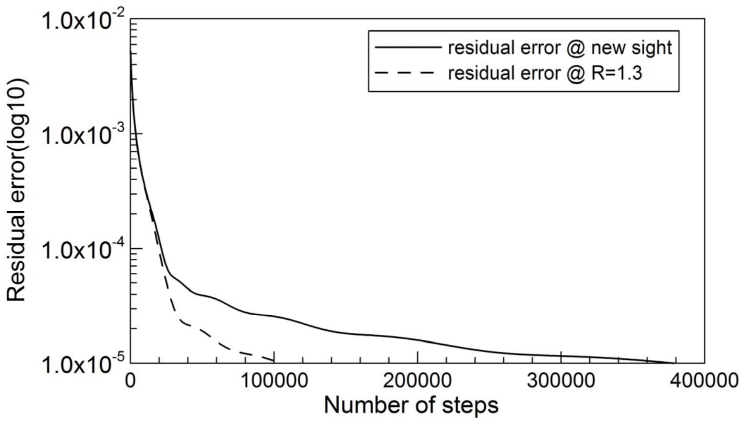


Figure 11: The convergence of residual error via number of steps with the new sight DQ and the scaling DQ with $R = 1.3$ for example 3.

Its boundary condition is $u(x,0) = e^x$, and the exact solution is

$$u(x,y) = \exp\left(x + \frac{y}{3}\right). \tag{26}$$

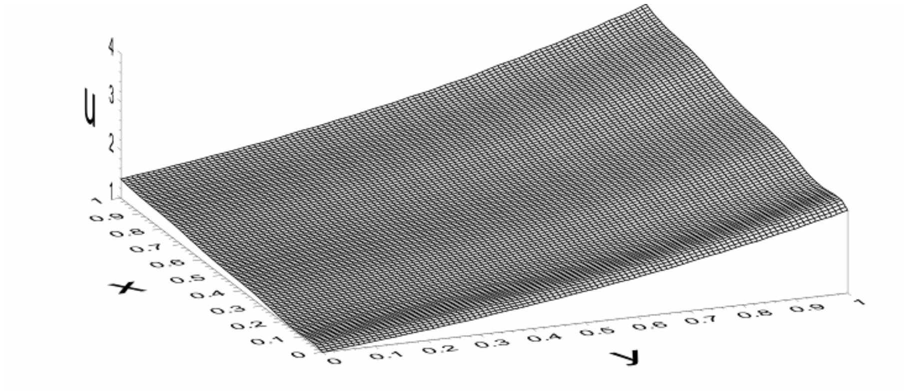
We select $n = 30$ and $m = 12$ in each variable of x and y. The scaling factor $R = 1.3$ and the parameters in FTIM are $h = 0.001$, $\nu = 0.1$ and $\varepsilon = 10^{-5}$. The initial guess is $u_{ij} = e^{x_i}$, where $i, j = 1, \ldots, n$. It converges up to 378502 steps when using the new DQ and 104866 steps when using the scaling DQ with $R = 1.3$. The convergence of residual error is shown in Fig. 11. The results and errors are respectively shown in Figs. 12(a)-12(c) and Figs. 13(a) -13(d).

The maximum errors are 5.171×10^{-2} in the new DQ and 4.201×10^{-2} in the scaling DQ with $R = 1.3$ when $x = 0.586$; 7.822×10^{-2} in the new DQ and 6.354×10^{-2} in the scaling DQ with $R = 1.3$ when $x = 1$; 2.972×10^{-2} in the new DQ and 1.775×10^{-2} in the scaling DQ with $R = 1.3$ when $y = 0.586$; 2.872×10^{-2} in the new DQ and 5.66×10^{-2} in the scaling DQ with $R = 1.3$ when $y = 1$. The errors in the new DQ are the same as those using a scaling factor in DQ.

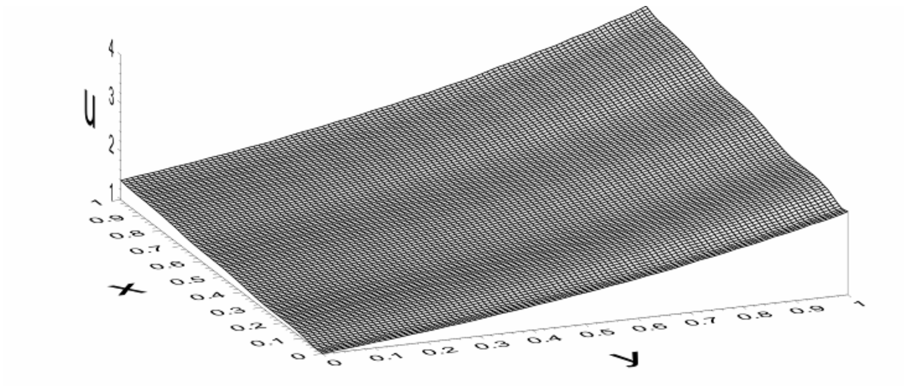
4.4 Example 4

In this example we also apply the new DQ and FTIM to solve a 2-D non-linear PDE with the governing equation:

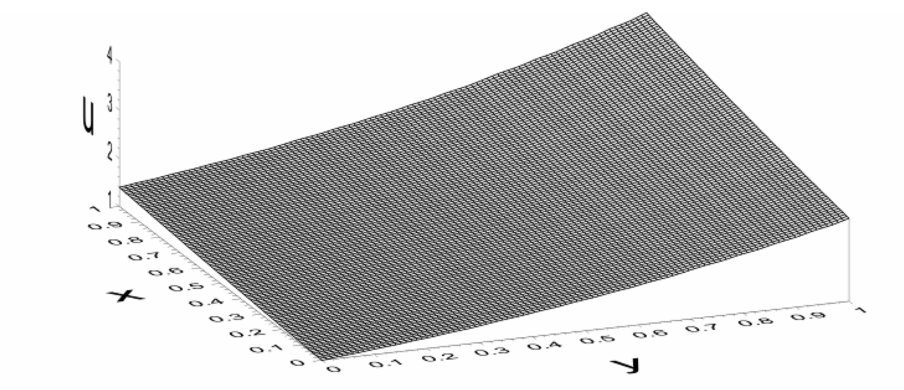
$$\frac{\partial u}{\partial x} + u\frac{\partial u}{\partial y} + u = 0, \;\; 0 \le x \le 1, \; 0 \le y \le 1. \tag{27}$$



(a) $m = 10$ in the new DQ.

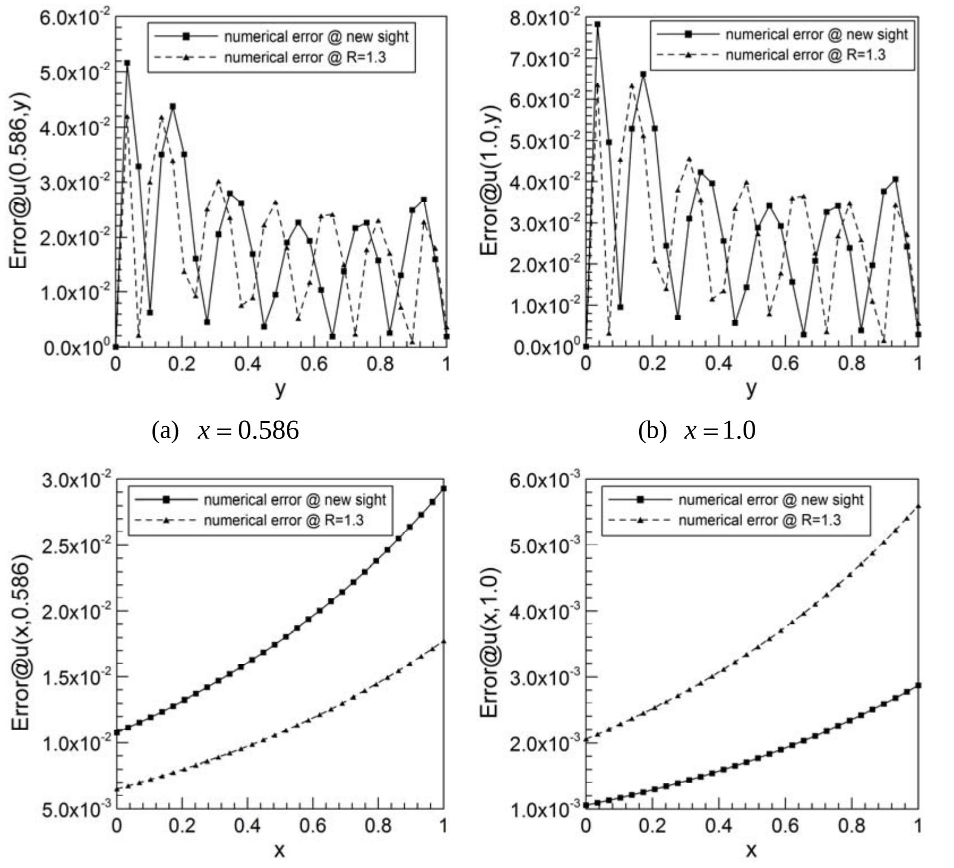


(b) $R = 1.3$ in the scaling DQ



(c) Exact solution

Figure 12: (a) Numerical solution in the new DQ, (b) numerical solution in the scaling DQ with $R = 1.3$ and (c) exact solution for example 3.



(a) $x = 0.586$ (b) $x = 1.0$

(c) $y = 0.586$ (d) $y = 1.0$

Figure 13: Comparing numerical errors in the new DQ and the scaling DQ with $R = 1.3$: (a) $x = 0.586$, (b) $x = 1$, (c) $y = 0.586$, and (d) $y = 1$ for example 3.

Its boundary condition is $u(0,y) = 1+y$, and its exact solution is

$$u(x,y) = \left(\frac{1+y}{2-e^{-x}}\right)e^{-x}. \tag{28}$$

The number of grid points is $n = 40$ and $m = 10$ in the new DQ. The scaling factor $R = 1.8$ and the parameters in FTIM are $h = 0.01$, $\nu = 1$ and $\varepsilon = 10^{-5}$. The initial guess is $u_{ij} = 1 + y_j$, where $i, j = 1, \ldots, n$. It converges within 3381 steps when using the new DQ and within 13418 steps when using the scaling DQ with $R = 1.8$. The convergence of residual error is shown in Fig. 14. The numerical results of

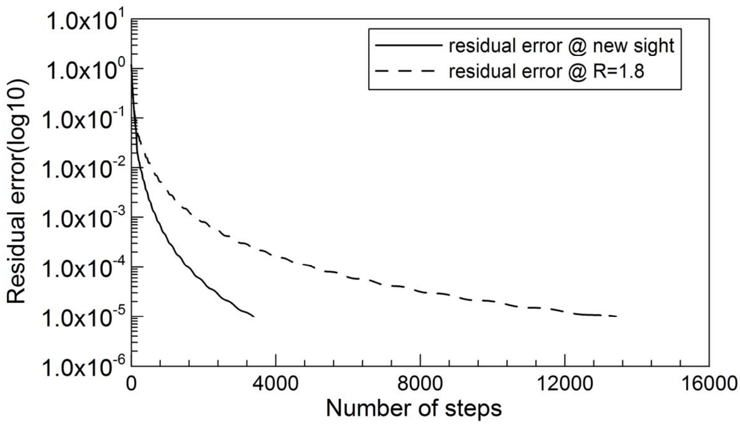


Figure 14: The convergence of residual error via number of steps with the new sight DQ and the scaling DQ with $R = 1.8$ for example 4.

the new DQ, the scaling DQ with $R = 1.8$ and the exact solution are enclosed in Figs. 15(a)-15(c). The errors are shown in Figs. 16(a)-16(d). The maximum errors are 5.759×10^{-5} in the new DQ and 1.294×10^{-4} in the scaling DQ with $R = 1.8$ when $x = 0.538$; 2.912×10^{-5} in the new DQ and 1.514×10^{-4} in the scaling DQ with $R = 1.8$ when $x = 1$; 1.144×10^{-4} in the new DQ and 3.665×10^{-4} in the scaling DQ with $R = 1.8$ when $y = 0.538$; 1.488×10^{-4} in the new DQ and 4.764×10^{-4} in the scaling DQ with $R = 1.8$ when $y = 1$.

4.5 Example 5

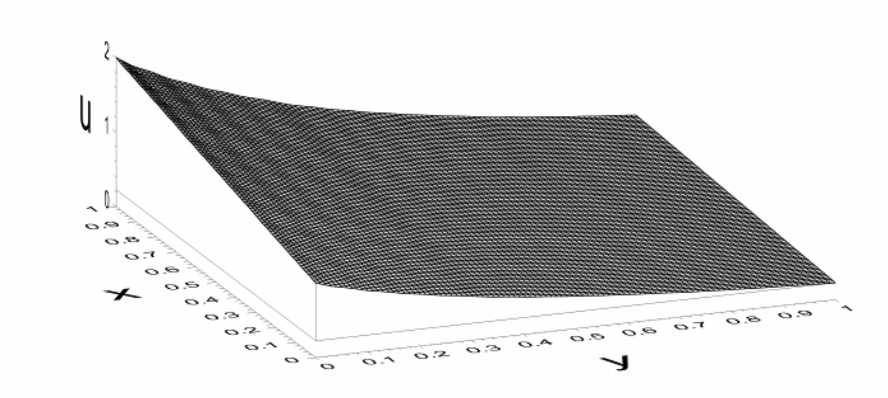
Liu (2008, 2009) was the first by applying the FTIM and a novel finite difference technique to treat the elliptic type PDEs in arbitrary plane domain. In this example we apply the new DQ and FTIM to solve the boundary value problem of a non-linear ellipitic type PDE [Atluri and Zhu (1998a, 1998b); Zhu, Zhang and Atluri (1998, 1999)]:

$$\Delta u(x,y) + \alpha^2 u(x,y) + \beta u(x,y) = p(x,y), \;\; 0 \le x \le 1, \; 0 \le y \le 1, \tag{29}$$

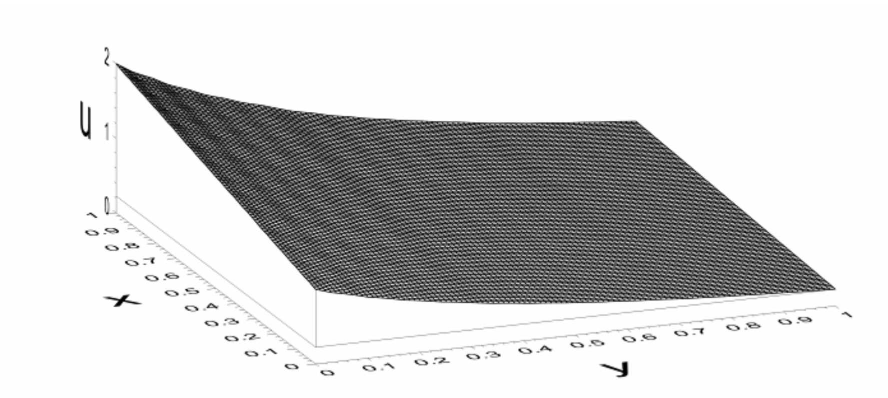
where we fix $\alpha = 1$ and $\beta = 0.001$. The exact solution is

$$u(x,y) = \frac{-5}{6}(x^3 + y^3) + 3(x^2 y + xy^2). \tag{30}$$

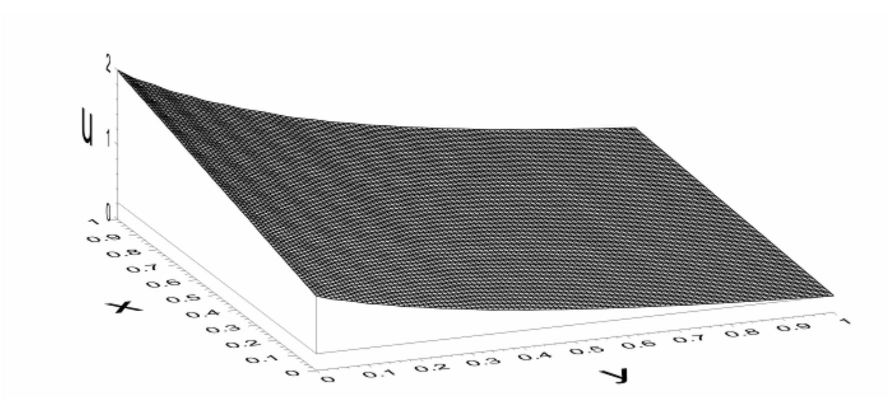
The exact $p(x,y)$ can be obtained by inserting the above $u(x,y)$ into Eq. (29).



(a) $m = 10$ in the new DQ.

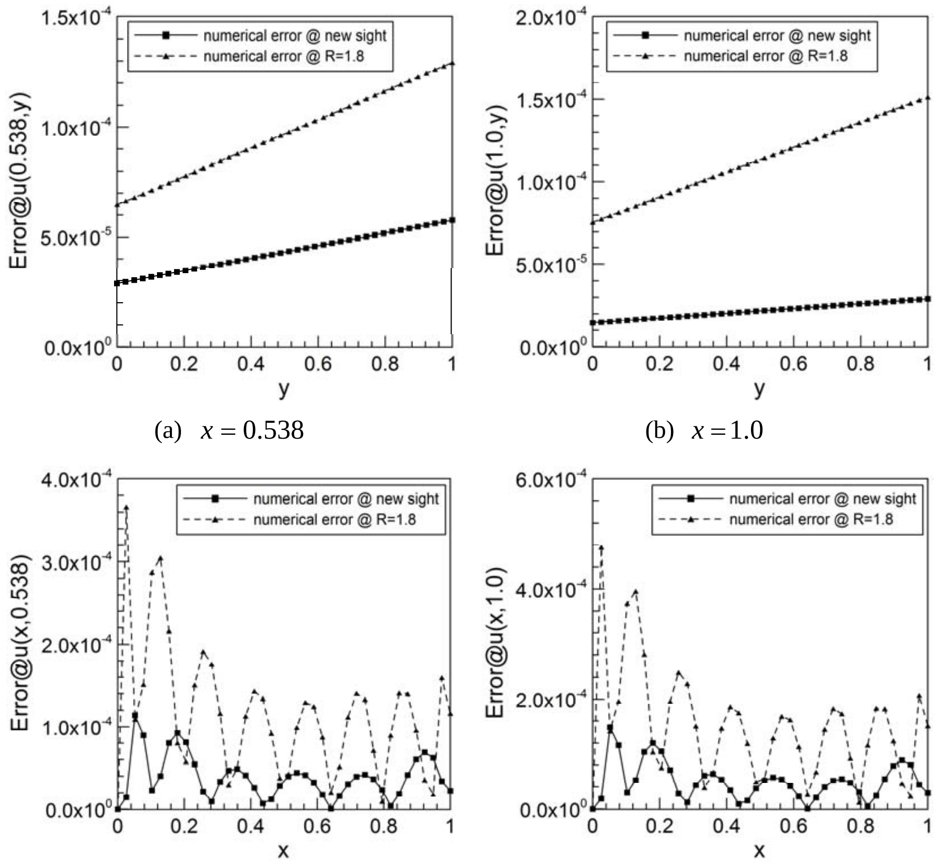


(b) $R = 1.8$ in the scaling DQ



(c) Exact solution

Figure 15: (a) Numerical solution in the new DQ, (b) numerical solution in the scaling DQ with $R = 1.8$ and (c) exact solution for example 4.



(a) $x = 0.538$ (b) $x = 1.0$

(c) $y = 0.538$ (d) $y = 1.0$

Figure 16: Comparing numerical errors in the new DQ and the scaling DQ with $R = 1.8$: (a) $x = 0.538$, (b) $x = 1$, (c) $y = 0.538$, and (d) $y = 1$ for example 4.

The number of grid points is $n = 30$ and $m = 12$ in new DQ. The scaling factor $R = 1.7$ and the parameters in FTIM are $h = 0.0005$, $\nu = -2$ and $\varepsilon = 10^{-5}$. The initial guess is $u_{ij} = -0.1$, where $i, j = 1, \ldots, n$. They converge within the same steps which are 614 steps when using the new DQ and the scaling DQ with $R = 1.7$. The convergence of residual error is shown in Fig. 17. The numerical results of the new DQ, the scaling DQ with $R = 1.7$ and the exact solution are enclosed in Figs. 18(a)-18(c). The errors are shown in Figs. 19(a)-19(d). The maximum errors are 4.555×10^{-5} in the new DQ and 4.556×10^{-5} in the scaling DQ with $R = 1.7$ when $x = 0.586$; 3.250×10^{-5} in the new DQ and 3.251×10^{-5} in the scaling DQ

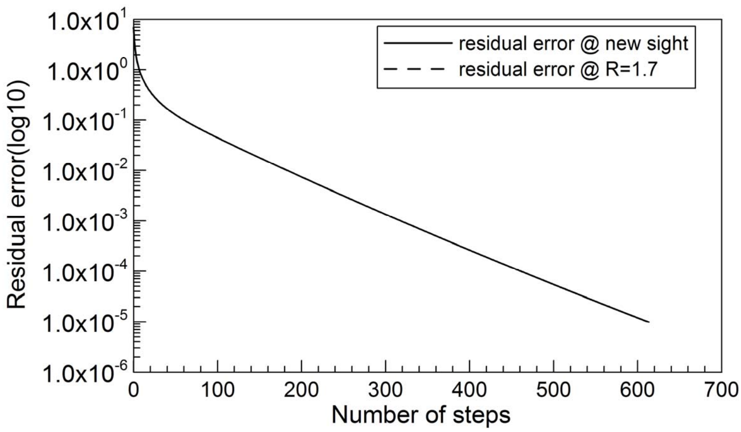
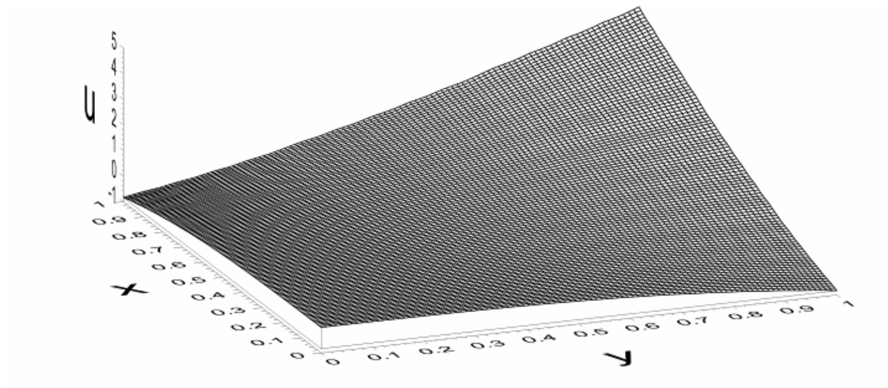


Figure 17: The convergence of residual error via number of steps with the new sight DQ and the scaling DQ with $R = 1.7$ for example 5.

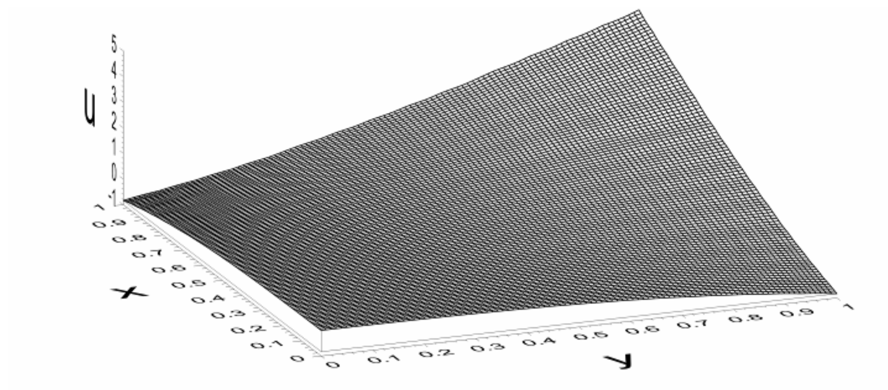
with $R = 1.7$ when $x = 0.759$; 4.555×10^{-5} in the new DQ and 4.556×10^{-5} in the scaling DQ with $R = 1.7$ when $y = 0.586$; 3.250×10^{-5} in the new DQ and 3.251×10^{-5} in the scaling DQ with $R = 1.7$ when $y = 0.759$. We can observe that the results in the new DQ and using the scaling factor in DQ are the same.

5 Conclusions

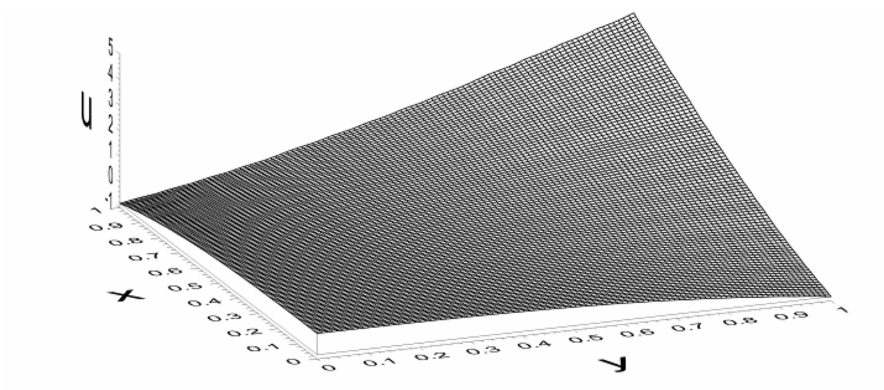
We have used $(m-1)$th order polynomial test functions by n grid points in a global differential quadrature formulation, of which the size of Vandermonde matrix is $m \times n$, and m is much less than n. We find that the $(m-1)$th order polynomial test functions are accurate enough to express the solutions, and the novel method significantly improves the ill-condition of algebraic equations. The size of the underdeterminate Vandermonde matrix is influenced by the number of grid points. Five examples were applied by using the new DQ and FTIM to calculate the solutions. Comparing with exact solutions, the numerical errors of the new DQ are very small, and the results are sometimes more accurate than those obtained from an improved DQ which uses the scaling factor. Consequently, the new DQ could be successfully applied in solving 2-D problems, especially in the elliptic type PDEs, for its easy numerical-implementation and time saving.



(a) $m = 12$ in the new DQ.

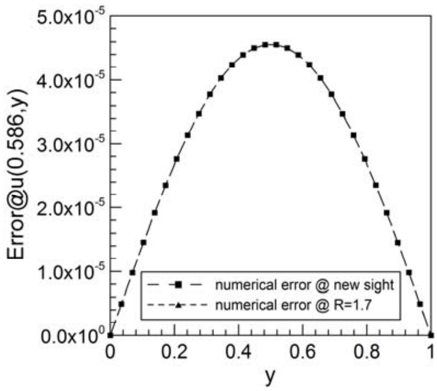


(b) $R = 1.7$ in the scaling DQ

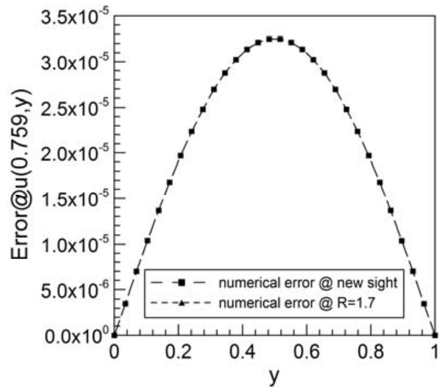


(c) Exact solution

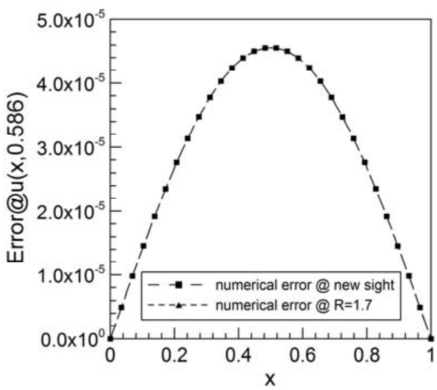
Figure 18: (a) Numerical solution in the new DQ, (b) numerical solution in the scaling DQ with $R = 1.7$ and (c) exact solution for example 5.



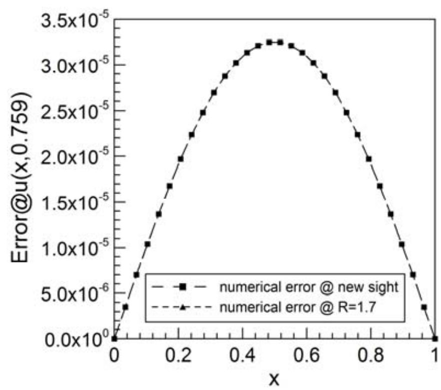
(a) $x = 0.586$



(b) $x = 0.759$



(c) $y = 0.586$



(d) $y = 0.759$

Figure 19: Comparing numerical errors in the new DQ and the scaling DQ with $R = 1.7$: (a) $x = 0.586$, (b) $x = 0.759$, (c) $y = 0.586$, and (d) $y = 0.759$ for example 5.

References

Atluri, S. N.; Zhu, T. L. (1998a): A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Comp. Mech.*, vol. 22, pp. 117-127.

Atluri, S. N.; Zhu, T. L. (1998b): A new meshless local Petrov-Galerkin (MLPG) approach to nonlinear problems in computer modeling and simulation. *Comp. Model. Simul. Eng.*, vol. 3, pp. 187-196.

Bellman, R. E.; Casti, J. (1971): Differential quadrature and long-term integration. *J. Math. Anal. Appl.*, vol. 34, pp. 235-238.

Bellman, R. E.; Kashef, B. G.; Casti, J. (1972): Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations. *J. Comp. Phys.*, vol. 10, pp. 40-52.

Liu, C.-S. (2008): A fictitious time integration method for two-dimensional quasilinear elliptic boundary value problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 33, pp. 179-198.

Liu, C.-S. (2009): A fictitious time integration method for a quasilinear elliptic boundary value problem, defined in arbitrary plane domain. *CMC: Computers, Materials & Continua*, vol. 11, pp. 15-32.

Liu, C.-S.; Atluri, S. N. (2008): A novel time integration method for solving a large system of non-linear algebraic equations. *CMES: Computer Modeling in Engineering & Sciences*, vol. 31, pp. 71-83.

Liu, C.-S.; Hong, H. K.; Atluri, S. N. (2010): Novel algorithms based on the conjugate gradient method for inverting ill-conditioned matrices, and a new regularization method to solve ill-posed linear systems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 60, pp. 279-308.

Quan, J. R.; Chang, C.-T. (1989a): New insights in solving distributed system equations by the quadrature methods-I. *Comput. Chem. Eng.*, vol. 13, pp. 779-788.

Quan, J. R.; Chang, C.-T. (1989b): New insights in solving distributed system equations by the quadrature methods-II. *Comput. Chem. Eng.*, vol. 13, pp. 1017-1024.

Shu, C. (2000): Differential Quadrature and its Applications in Engineering. Springer, Berlin.

Zhu, T.; Zhang, J.; Atluri, S. N. (1998): A meshless local boundary integral equation (LBIE) method for solving nonlinear problems. *Comp. Mech.*, vol. 22, pp. 174-186.

Zhu, T.; Zhang, J.; Atluri, S. N. (1999): A meshless numerical method based on the local boundary integral equation (LBIE) to solve linear and non-linear boundary value problems. *Eng. Anal. Bound. Elem.*, vol. 23, pp. 375-389.

Zong, Z.; Lam, K. Y. (2002): A localized differential quadrature (LDQ) method and its application to the 2D wave equation. *Comput. Mech.*, vol. 29, pp. 382-391.