International Journal of
Mathematical Modelling & Computations
Vol. 07, No. 01, Winter 2015, 67- 77

Using Chebyshev polynomials zeros as point grid for numerical solution of linear and nonlinear PDEs by differential quadrature-based radial basis functions

S. Kosari [a,*] and M. Erfanian[b]

[a,b] *Department of Mathematics, University of Zabol, Zabol, Iran.*

Abstract. Radial Basis Functions (RBFs) have been found to be widely successful for the interpolation of scattered data over the last several decades. The numerical solution of nonlinear Partial Differential Equations (PDEs) plays a prominent role in numerical weather forecasting, and many other areas of physics, engineering, and biology. In this paper, Differential Quadrature (DQ) method- based RBFs are applied to find the numerical solution of the linear and nonlinear PDEs. The multiquadric (MQ) RBFs as basis function will introduce and applied to discretize PDEs. Differential quadrature will introduce briefly and then we obtain the numerical solution of the PDEs. DQ is a numerical method for approximate and discretized partial derivatives of solution function. The key idea in DQ method is that any derivatives of unknown solution function at a mesh point can be approximated by weighted linear sum of all the functional values along a mesh line.

Received: 7 September 2016, Revised: 11 December 2016, Accepted: 13 February 2017.

Keywords: radial basis function, differential quadrature, PDE, collocation method.

Index to information contained in this paper

1. Introduction

Since the radial basis functions (RBFs) method is a meshfree method which is very easy to used, so many of researchers are very interested to work on it. RBFs are a powerful tool in interpolation multivariable functions or approximation solution of partial or ordinary differential equations. RBFs have really a meshfree nature. We

*Corresponding author. Email: sajad.kosari@uoz.a.ir

Table 1. most popular RBFs

RBF Name	Formula
Multiquadric	$\varphi(r_j) = \sqrt{r_j^2 + c^2}$
Inverse multiquadric	$\varphi(r_j) = \frac{1}{\sqrt{r_j^2+c^2}}$
Thin plate spline	$\varphi(r_j) = r_j^2 \ln(r_j^2 + c^2)$
Gaussian	$\varphi(r_j) = e^{-cr_j^2}$

introduce RBF in this paper. Some of RBFs has a shape parameter c and we called them parametric RBFs. Parametric RBFs have very interested properties, they are smooth and infinitely differentiable. In interpolation or solving PDEs numerically, their system matrix is nonsingular and thus interpolation problem with RBFs or problem of solving PDE with RBFs has a unique solution and this is great.
It is well known that the value of c strongly influences the accuracy of approximation, which is used to approximate the solution of PDEs. Thus, there exists a problem of how to select a "good" value of c so that the numerical solution of PDEs can achieve satisfactory accuracy. In general, there are three main factors that could affect the optimal shape parameter c for giving the most accurate results. These three factors are the scale of supporting region, the number of supporting nodes, and the distribution of supporting nodes [1]. Among the three factors, the effect of nodes distribution is the most difficult to be studied since there are infinite kinds of distribution [1]. Of course Shu and his associates completely investigated this problem in [1].
We Assume that f be a real value function that defined on the real line R, then the function $\varphi : R^d \to R$ that $\varphi(r_j) = f(r_j)$ and $r_j = ||x - x_j||$ where $x, x_j \in R^d$ is said a radial function. $||.||$ Is the Euclidian norm and x_j is a special mesh point and called the center of radial function. List of most popular RBFs are shown in Table 1. We note that again $r_j = ||x - x_j||$
RBFs are usually divide into two categories: globally supported and locally supported RBFs. we say, the RBF is called locally supported if $\lim_{r\to\infty} \varphi\ (r) = 0$, and called globally supported if $\lim_{r\to\infty} \varphi\ (r) = \infty$. Multiquadric and thin plat spline are globally supported and inverse multiquadric and Gaussian are the locally supported RBFs. In this paper, we are interested to used globally supported and in particular MQ. The MQ method was originally used for interpolation of scattered data. MQ is a RBF that was introduced by Hardy for multivariate data interpolation [2]. Franke tested this and other functions, and found that multiquadric yielded faster convergence than other radial basis functions [3]. The use of MQ as a basis function in the global collocation method for solving PDEs was proposed by Kansa [4], the free parameter c in MQ is known as the shape parameter because its value affects the shape of the function [5]. The accuracy of the global collocation method could be increased by increasing the value of the shape parameter [5]. When the shape parameter is too large, however, round- off error dominates, and the method loses its accuracy [5].

2. The description of Differential Quadrature (DQ) method

The differential quadrature (DQ) method was introduced by Richard Bellman and his associates in the early of 1970s [6], [7]. The basic idea of the DQ method is that any derivative at a mesh point can be approximated by a weighted linear sum of all the functional values along a mesh line [7]. Currently, the DQ method has been extensively applied in engineering for the rapid and accurate solution of various

linear and nonlinear differential equations [8], [9], [10]. Differential quadrature (DQ) method is a numerical method for solving partial or ordinary differential equations. In this method, we approximate the spatial derivatives of the function f at mesh points $x_i \in R^d$ using linear weighted sum of all the functional values at points in the domain of the problem. We assume N grid points on the real axis with step length. In the case $d = 2$, The discretization of the nth and the mth order derivatives by DQ method at a point (x_i,y_i) with respect to x and y, respectively, is given by equations (1) and (2) that $f_x^{(n)}$ is nth order derivative of f with respect to x and $f_y^{(m)}$is mth order derivative of f with respect to y. however we will have the following equations

$$f_x^{(n)}(x_i, y_i) = \sum_{j=1}^{N} w_{ij}^{(n)} f(x_j, y_j), \qquad i = 1, 2, \ldots, N, \tag{1}$$

$$f_y^{(m)}(x_i, y_i) = \sum_{j=1}^{N} v_{ij}^{(m)} f(x_j, y_j), \qquad i = 1, 2, \ldots, N. \tag{2}$$

Now, for the case of dependent time PDEs, if t be the temporal variable , then the equations of DQ discretization for the first and for the second order derivatives of a univariable function are as below

$$f'(x_i, t) = \sum_{j=1}^{N} w_{ij} f(x_j, t) \qquad , \qquad i = 1, 2, \ldots, N \tag{3}$$

$$f''(x_i, t) = \sum_{j=1}^{N} v_{ij} f(x_j, t) \qquad , \qquad i = 1, 2, \ldots, N \tag{4}$$

Where w_{ij} and v_{ij} are unknown and we called them the weighting coefficients of the derivatives of first and second order. There are many approaches to find these coefficients such as Bellmans approaches [11], Ram Jiwari in [6] and Shus approach in [8]. From these approaches, Shus approach is very general approach in the recent years. The function $f(x, y)$ in (3) and (4) are called test functions and for obtain the weighting coefficients we need a suitable test function. Some of the most general test functions are: Legendre polynomials, Lagrange interpolation polynomials, Lagrange interpolated cosine and Radial Basis Functions. We are interested that use RBFs and in particular Multiquadric (MQ) as test functions in this paper. However in 2-dimansion case, for obtaining the coefficients w_{ij} and v_{ij} we substitute the function MQ with equation

$$\varphi_k(x, y) = \sqrt{|x - x_k|^2 + |y - y_k|^2 + c^2},$$

in the equations (1) and (2) and obtain the below equations [7].

$$\begin{cases} \varphi_{kx}^{(n)}(x_i, y_i) = \sum_{j=1}^{N} w_{ij}^{(n)} \varphi_k(x_j, y_j), & i = 1, 2, \ldots, N, \quad i \neq k, \\ \sum_{j=1}^{N} w_{ij}^{(n)} = 0 & \qquad\qquad\quad i = k. \end{cases} \tag{5}$$

$$\begin{cases} \varphi_{ky}^{(m)}(x_i, y_i) = \sum_{j=1}^{N} v_{ij}^{(m)} \varphi_k(x_j, y_j), & i = 1, 2, \ldots, N, \quad i \neq k, \\ \sum_{j=1}^{N} v_{ij}^{(m)} = 0 & i = k. \end{cases} \tag{6}$$

That $\varphi_{kx}^{(n)}$ and $\varphi_{ky}^{(m)}$ are nth and mth order of derivative of φ_k with respect to x and y respectively. For the given i, any of equation systems of (5) and (6) has N unknowns with N equations. So, with solving this equation system, we can obtain the weighting coefficients. In section five, we will describe completely how to solve (5) and (6) for any i and obtain the coefficients. Now, note that, we can see easily that, the first and second order derivatives of MQ are as below:

$$\varphi_{kx}^{(1)}(x, y) = \frac{x - x_k}{\sqrt{(x - x_k)^2 + (y - y_k)^2 + c^2}} \tag{7}$$

$$\varphi_{kx}^{(2)}(x, y) = \frac{(y - y_k)^2 + c^2}{((x - x_k)^2 + (y - y_k)^2 + c^2)^{\frac{3}{2}}} \tag{8}$$

That, in above, $\varphi_{kx}^{(1)}$ and $\varphi_{kx}^{(2)}$ are the first and the second order derivative of φ_k with respect to x respectively. Off curse, the first and the second order derivative of φ_k with respect to y will obtained as below

$$\varphi_{ky}^{(1)}(x, y) = \frac{y - y_k}{\sqrt{(x - x_k)^2 + (y - y_k)^2 + c^2}} \tag{9}$$

$$\varphi_{ky}^{(2)}(x, y) = \frac{(x - x_k)^2 + c^2}{((x - x_k)^2 + (y - y_k)^2 + c^2)^{\frac{3}{2}}} \tag{10}$$

That, in above, $\varphi_{ky}^{(1)}$ and $\varphi_{ky}^{(2)}$ are the first and the second order derivative of φ_k with respect to y respectively.

3. How we select the nodes

It is well known that the position of the nodes is a very important subject in interpolation or numerical solution of PDE problems. The accuracy and convergences speed are depended on nodes position. For selecting the nodes $x_i \in R^d$ there are many methods, uniform grid, random grid and the Chebyshev-Gauss-Lobatto point's grid. But in this paper we used zeros of Chebyshev polynomials as nodes $x_i \in R^d$. It is well known that, in problem of interpolation of a real value function, interpolate function has minimum value of error when the nodes have been the zeros of Chebyshev polynomials, we will introduce and describe the Chebyshev polynomial briefly in the next section, and we will used the Chebyshev polynomials zeros as node grid for numerical solution of PDEs with DQ- based RBFs method.

4. Chebyshev polynomials

Chebyshev polynomials defined on $[0, 1]$ and denoted by $T_n(x)$ and represented as below:

$$T_n(x) = \cos(n \arccos x) \tag{11}$$

If we let $x = \cos\theta$ then we have $\theta = \arccos x$, so, (11) will rewriting as below

$$T_n(x) = \cos n\theta \tag{12}$$

From the

$$\cos((n+1)\theta) + \cos((n-1)\theta) = 2\cos\theta\cos n\theta$$

And with use of the (11) and (12) we have

$$T_{n+1}(x) + T_{n-1}(x) = 2xT_n(x)$$

Or we can say

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \tag{13}$$

From the iterative formulae of (13) and with note that

$$T_0(x) = 1 \qquad\qquad T_1(x) = x$$

That obtain from (11) directly, we can obtain all of the $T_n(x)$ formulation. Here we present some of Chebyshev polynomials:

$$\begin{gathered} T_2(x) = 2x^2 - 1 \\ T_3(x) = 4x^3 - 3x \\ T_4(x) = 8x^4 - 8x^2 + 1 \end{gathered}$$

For obtaining zeros of Chebyshev polynomials we first note that x is called a zero of $T_n(x)$ if we have

$$T_n(x) = \cos n\theta = 0 \qquad \Longrightarrow \qquad n\theta = k\pi + \frac{\pi}{2} = (2k+1)\frac{\pi}{2}$$

So we obtain the below formulae

$$\theta = \frac{(2k+1)\pi}{2n}$$

And because of $x = \cos\theta$ then the zeros of $T_n(x)$ will obtained as following iterative formulae

$$x_k = \cos\frac{(2k+1)\pi}{2n} \qquad , \qquad k = 0, 1, 2, \cdots, n-1$$

5. Determining weighting coefficients

In this section, for 2-dimansion's case, we describe how obtain the weighting coefficients $w_{ij}^{(1)}$,$w_{ij}^{(2)}$, $v_{ij}^{(1)}$ and $v_{ij}^{(2)}$. First we note that $\{x_i\}_{i=1}^{N}$ are zeros of Chebyshev polynomial of degree N, now for every $i = 1, 2, \cdots, N$ we set $x_i = y_i$. Then the set of $\{(x_i, y_j)\}$ that $i, j = 1, 2, \cdots, N$ are all of the nodes which we discretized the PDEs in this points. We can see easily that we have N^2 nodes. If we denote this N^2 nodes as $\{(x_k, y_k)\}_{k=1}^{N^2}$ then we can easily see that $k = i + (j-1)N$ that $i, j = 1, 2, \cdots, N$. Of course, these unknown coefficients will obtain from (5) and (6) as follow. We just describe how determining $w_{ij}^{(1)}$, and then $w_{ij}^{(2)}$, $v_{ij}^{(1)}$ and $v_{ij}^{(2)}$ will obtained completely in a similar manner. If we extend the (5) for $n = 1$ and for $M = N^2$ nodes (x_i, y_i), $i = 1, 2, \cdots, M$, then for any k and $i = 1, 2, \cdots, M$ we have M equation with M unknown $w_{ij}^{(1)}$ that $i, j = 1, 2, \cdots, M$ However, with solution of this M equation with M unknown $w_{ij}^{(1)}$ we determine all of the arrays of ith row of below matrix:

$$W^{(1)} = \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & \dots & w_{1M}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & \dots & w_{2M}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M1}^{(1)} & w_{M2}^{(1)} & \dots & w_{MM}^{(1)} \end{bmatrix}.$$

The ith row of $W^{(1)}$ is the solution of the following linear system of equations:

$$\begin{cases} w_{i1}^{(1)}\varphi_{11} + w_{i2}^{(1)}\varphi_{12} + \cdots + w_{iM}^{(1)}\varphi_{1M} = \varphi_{1i}^{(1)}, \\ w_{i1}^{(1)}\varphi_{21} + w_{i2}^{(1)}\varphi_{22} + \cdots + w_{iM}^{(1)}\varphi_{2M} = \varphi_{2i}^{(1)}, \\ \vdots \\ w_{i1}^{(1)}\varphi_{(i-1)1} + w_{i2}^{(1)}\varphi_{(i-1)2} + \cdots + w_{iM}^{(1)}\varphi_{(i-1)M} = \varphi_{(i-1)i}^{(1)}, \\ w_{i1}^{(1)} + w_{i2}^{(1)} + \cdots + w_{iM}^{(1)} = 0, \\ w_{i1}^{(1)}\varphi_{(i+1)1} + w_{i2}^{(1)}\varphi_{(i+1)2} + \cdots + w_{iM}^{(1)}\varphi_{(i+1)M} = \varphi_{(i+1)i}^{(1)}, \\ \vdots \\ w_{i1}^{(1)}\varphi_{M1} + w_{i2}^{(1)}\varphi_{M2} + \cdots + w_{iM}^{(1)}\varphi_{MM} = \varphi_{Mi}^{(1)}. \end{cases} \tag{14}$$

And in matrix form we can rewrite the system of (14) as follow:

$$\Phi W_i^{(1)} = \varphi^{(1)}.$$

That

$$\Phi = \begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1M} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{(i-1)1} & \varphi_{(i-1)2} & \cdots & \varphi_{(i-1)M} \\ 1 & 1 & \cdots & 1 \\ \varphi_{(i+1)1} & \varphi_{(i+1)2} & \cdots & \varphi_{(i+1)M} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{M1} & \varphi_{M2} & \cdots & \varphi_{MM} \end{bmatrix},$$

and

$$W_i^{(1)} = [w_{i1}^{(1)}, w_{i2}^{(1)}, \cdots, w_{iM}^{(1)}]^T,$$

and

$$\varphi^{(1)} = [\varphi_{1i}^{(1)}, \varphi_{2i}^{(1)}, \cdots, \varphi_{Mi}^{(1)}]^T.$$

That φ_{ij} is multiquadric RBF and defined as

$$\varphi_{ij} = \varphi_i(x_j, y_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + c^2}.$$

Also, $\varphi_{ij}^{(1)}$ is the first order derivative of φ_i with respect to x in (x_j, y_j). We did obtained the first order derivative of multiquadric in (7). However with solution of linear system of equations in (14) we get the unknown coefficients $w_{ij}^{(1)}$.

6. Numerical examples

Example 6.1 in this section we explain how employ our method for solving PDEs. Consider the 2-dimansion Poisson equation in a square domain $[-1, 1] \times [-1, 1]$ that we getting it from [7].

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -2\pi^2 \sin \pi x \sin \pi y. \tag{15}$$

Now we discretized the above PDE as below

$$\frac{\partial^2 u(x_i, y_i)}{\partial x^2} + \frac{\partial^2 u(x_i, y_i)}{\partial y^2} = -2\pi^2 \sin \pi x_i \sin \pi y_i \quad , \quad i = 1, 2, \cdots, M.$$

That $M = N^2$ and then from the concept of Differential Quadrature method we obtain the following equations

$$\sum_{j=1}^{M} w_{ij}^{(2)} u(x_j, y_j) + \sum_{j=1}^{M} v_{ij}^{(2)} u(x_j, y_j) = -2\pi^2 \sin \pi x_i \sin \pi y_i \quad , \quad i = 1, 2, \cdots, M. \tag{16}$$

In equation system (16), we note that the coefficients $w_{ij}^{(2)}$ and $v_{ij}^{(2)}$ were obtained before, from (5) and (6), and indeed they are not unknown. The equation

Table 2. Relative error for a linear problem (Example 6.1)

Number of nodes	L_2 error $\times 10^{-2}$	Optimal shape parameter c	CPU times (s)
4	0.0814	1.524	0.026
16	0.0103	1.216	0.051
36	0.0018	1.143	0.083
49	0.00748	1.029	0.110
64	0.00817	0.935	0.184
81	0.02573	0.729	0.223

(16) is a linear system of equations that includes M equations and M unknown $\{u(x_i, y_i)\}_{i=1}^{M}$ We can easily rewriting (16) as below

$$\sum_{j=1}^{M} (w_{ij}^{(2)} + v_{ij}^{(2)}) u(x_j, y_j) = -2\pi^2 \sin \pi x_i \sin \pi y_i \quad , \quad i = 1, 2, \cdots, M. \tag{17}$$

And then in matrix form, the system of equation of (17) rewriting as below

$$AU = B. \tag{18}$$

That in (18) we have

$$A = \begin{bmatrix} w_{11}^{(2)} + v_{11}^{(2)} & w_{12}^{(2)} + v_{12}^{(2)} & \dots & w_{1M}^{(2)} + v_{1M}^{(2)} \\ w_{21}^{(2)} + v_{21}^{(2)} & w_{22}^{(2)} + v_{22}^{(2)} & \dots & w_{2M}^{(2)} + v_{2M}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M1}^{(2)} + v_{M1}^{(2)} & w_{M2}^{(2)} + v_{M2}^{(2)} & \dots & w_{MM}^{(2)} + v_{MM}^{(2)} \end{bmatrix}_{M \times M}.$$

$$U = [u(x_1, y_1), u(x_2, y_2), \dots, u(x_M, y_M)]^T,$$

$$B = -2\pi^2 [\sin(\pi x_1) \sin(\pi y_1), \sin(\pi x_2) \sin(\pi y_2), \dots, \sin(\pi x_M) \sin(\pi y_M)].$$

That in above notations, T is representing the transpose of a matrix or a vector. With solution of (17) we obtain the numerical solution of PDE in (15). For obtaining the numerical results for the *Example 6.1* by proposed method, we used MATLAB software and get the results in Table 2.
Now, to clearly show the behavior of the RBF- based DQ method for the *Example 6.1*, the relative errors are plotted against shape parameter c for various nodes and optimal shape parameters c are plotted against number of nodes. This can be observed from Figure1.

Example 6.2 Now we solve a nonlinear PDE with this method, consider the below nonlinear PDE that we get that from [7]. We suppose that $[-1, 1] \times [-1, 1]$ be the domain of this problem.

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + u(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}) - 2(x + y)u = 4, \tag{19}$$

with the below Dirichlet boundary condition for the four edges of the square domain

$$\begin{cases} u(x=-1)=1+y^2, & u(x=1)=1+y^2, \\ u(y=-1)=1+x^2, & u(y=1)=1+x^2. \end{cases} \tag{20}$$

We can see easily that the exact solution of this problem is $u(x,y)=x^2+y^2$. This example is very different with first example, because this problem is nonlinear and hence the system of equations that we obtain from discretization of (19) is nonlinear and indeed it is not easy to solve it. If we assume that sequence $\{x_i\}_{i=1}^{N}$ be the zeros of Chebyshev polynomial from degree N, then for discretization of (19) we assume that the sequence $\{(x_k,y_k)\}_{k=1}^{N^2}$ be the grid points where in that we have $k=i+(j-1)N$ where $i,j=1,2,\cdots,N$. However, with the above assumptions, we have discretized the equation (19) as follow

$$\frac{\partial^2 u(x_i,y_i)}{\partial x^2}+\frac{\partial^2 u(x_i,y_i)}{\partial y^2}+u(x_i,y_i)(\frac{\partial u(x_i,y_i)}{\partial x}+\frac{\partial u(x_i,y_i)}{\partial y})-2(x_i+y_i)u(x_i,y_i)=4. \tag{21}$$

That $i=1,2,\cdots,M=N^2$. Now with applying DQ method we have the following equations

$$\sum_{j=1}^{M} w_{ij}^{(2)}u(x_j,y_j)+\sum_{j=1}^{M} v_{ij}^{(2)}u(x_j,y_j)+u(x_i,y_i)[\sum_{j=1}^{M} w_{ij}^{(1)}u(x_j,y_j)$$

$$+\sum_{j=1}^{M} v_{ij}^{(1)}u(x_j,y_j)]-2(x_i+y_i)u(x_i,y_i)=4 \quad , \quad i=1,2,\cdots,M.$$

Or

$$\sum_{j=1}^{M}(w_{ij}^{(2)}+v_{ij}^{(2)})u(x_j,y_j)+u(x_i,y_i)\sum_{j=1}^{M}(w_{ij}^{(1)}+v_{ij}^{(1)})u(x_j,y_j)-2(x_i+y_i)u(x_i,y_i)=4. \tag{22}$$

If we set $a_{ij}=w_{ij}^{(2)}+v_{ij}^{(2)}$ and $b_{ij}=w_{ij}^{(1)}+v_{ij}^{(1)}$ then we can rewriting (22) as below

$$\sum_{j=1}^{M} a_{ij}u_j+u_i\sum_{j=1}^{M} b_{ij}u_j-2(x_i+y_i)u_i=4. \tag{23}$$

That in (23) we have $u_k=u(x_k,y_k)$ for $k=1,2,\cdots,M$ However, (23) is a nonlinear system of equations and we will solved it with Jacobi iteration method. First, we know ith equation of (23) is as follow

$$\sum_{j=1,j\neq i+1}^{M} a_{ij}u_j+a_{i(i+1)}u_{i+1}+u_i\sum_{j=1,j\neq i+1}^{M} b_{ij}u_j+b_{i(i+1)}u_iu_{i+1}-2(x_i+y_i)u_i=4.$$

And we have easily

$$u_{i+1}=\frac{1}{a_{i(i+1)}+b_{i(i+1)}u_i}\{4+2(x_i+y_i)u_i-\sum_{j=1,j\neq i+1}^{M} a_{ij}u_j-u_i\sum_{j=1,j\neq i+1}^{M} b_{ij}u_j\}.$$

Table 3. Relative error for a linear problem (Example 6.2)

Number of nodes	L_2 error $\times 10^{-2}$	Optimal shape parameter c	CPU times (s)
4	0.1584	1.524	1.227
16	0.0519	1.216	2.008
36	0.0053	1.143	2.109
49	0.0129	1.029	2.675
64	0.0417	0.935	3.013
81	0.0698	0.729	3.158

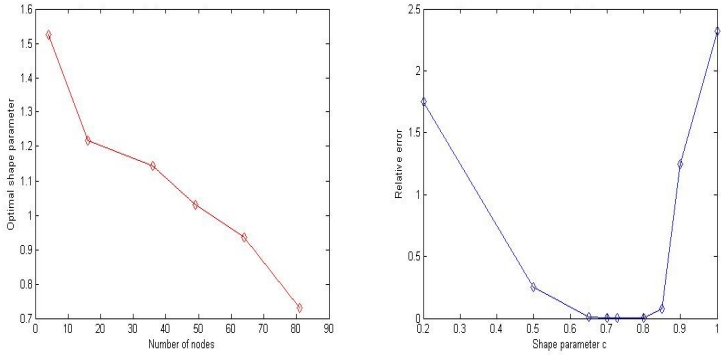

Figure 1. Optimal shape parameter against number of nodes in the left and relative error against shape parameter in the right.

In above we have $i = 1, 2, \cdots, M-1$. Until now, we obtain $u_2, u_3, \cdots, u_M$, now for obtaining u_1 from Mth equation of (23) we get the following equation

$$u_1 = \frac{1}{a_{M1} + b_{M1}u_M}\{4 + 2(x_M + y_M)u_M - \sum_{j=2}^{M} a_{Mj}u_j - u_M \sum_{j=2}^{M} b_{Mj}u_M\}.$$

Now, with an initial approximation $[u_1^{(0)}, u_2^{(0)}, \cdots, u_M^{(0)}]^T$, we obtain the solution of PDE (19) in (x_k, y_k), $k = 1, 2, \cdots, M$ from the following iterative equations

$$u_{i+1}^{(n+1)} = \frac{1}{a_{i(i+1)} + b_{i(i+1)}u_i^{(n)}}\{4+2(x_i+y_i)u_i^{(n)} - \sum_{j=1, j\neq i+1}^{M} a_{ij}u_j^{(n)} - u_i^{(n)} \sum_{j=1, j\neq i+1}^{M} b_{ij}u_j^{(n)}\},$$

and

$$u_1^{(n+1)} = \frac{1}{a_{M1} + b_{M1}u_M^{(n)}}\{4 + 2(x_M + y_M)u_M^{(n)} - \sum_{j=2}^{M} a_{Mj}u_j^{(n)} - u_M^{(n)} \sum_{j=2}^{M} b_{Mj}u_M^{(n)}\}.$$

That in above $u_k^{(n)}$ is approximation of exact solution $u(x_k, y_k)$ in step n of iteration. For obtaining the numerical results for the *Example 6.2* by proposed method, we used MATLAB software and get the following results that shownin Table 3.

Now, to clearly show the behavior of the RBF- based DQ method for the nonlinear example (*Example 6.2*), the relative errors are plotted against shape parameter c for various nodes and optimal shape parameters c are plotted against number of nodes. This can be observed from Figure 2.

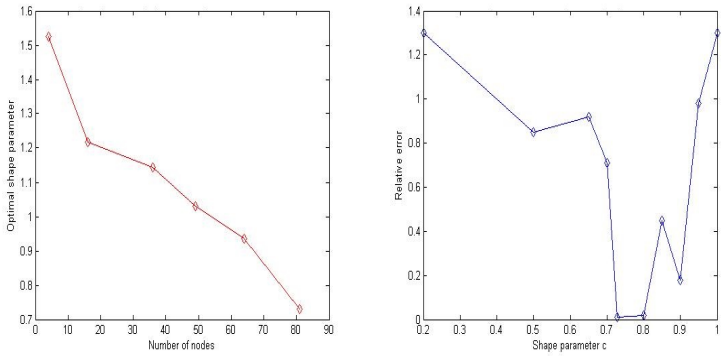

Figure 2. Optimal shape parameter against number of nodes in the left and relative error against shape parameter in the right.

7. Conclusions

From the numerical results in Tables 2 and 3, we see that the optimal shape parameter c where we obtained in the linear problem are same with those that we obtained in nonlinear case. We may expect that, for a fixed mesh points distribution, the optimal shape parameter c in the RBF- based DQ method remains the same for various problems. This phenomenon has also been observed by Shu [7]. It can be seen from tables 2 and 3 that the accuracy of numerical results can be improved by increasing the number of nodes, but when the number of nodes is further increased after a critical value, the accuracy of numerical results is decreased, because , when the number of nodes is inceased, the condition number of the matrix becomes very large and the equations system tend to be ill- conditioned.

References

[1] C. Shu, H. Ding, K.S. Yeo. Local radial basis function-based differentialquadrature method and its application to solve two-dimensional incompressible Navier–Stokes equations. Comput. Methods Appl. Mech. Engrg. 192 (2003) 941–954

[2] R. L. Hardy, multiquadric equation of topograghy and other irregular surfaces, journal of geophysical research 76 (1971) 1905-1915.

[3] R. Franke, scattered data interpolation: test of some methods, mathematics of computation 38 (57) (1982) 181- 200.

[4] E. J. Kansa, Multiquadric – A scattered data approximation scheme with application to computational fluid dynamic-II. Computers. Math. Applic. 19(8/9) , (1990) 147- 161

[5] S. Chantasiriwan, solution to harmonic and biharmonic problems with boundary conditions by collocation methods using multiquadrics as basis functions, international communication in heat and mass transfer 34 (2007) 313 – 320.

[6] RE. Bellman, J. Casti, Differential quadrature and long-term integration. J Math Anal Appl 1971;34:235–8.

[7] C. Shu H. Ding K.S. Yeo. Solution of partial differential equations by a global radial basisfunction-based differential quadrature method. Engineering Analysis with Boundary Elements 28 (2004) 1217–1226

[8] Shu C. Differential quadrature and its application in engineering. Great Britain: Athenaeum Press Ltd.; 2000.

[9] C. Shu, YT. Chew, Fourier expansion-based differential quadrature and its application to Helmholtz eigenvalue problems. Commun Numer Meth Engng 1997;13:643–53.

[10] CW. Bert, SK. Jang, AG. Striz, Two new methods for analyzing free vibration of structure components. AIAA J 1988;26:612–8.

[11] R. Bellman, BG. Kashef, J. Casti, Differential quadrature: a technique for the rapid solution of nonlinear partial differentialequations. J Comput Phys 1972;10(1):40–52.