Solution of partial differential equations by a global radial basis function-based differential quadrature method

C. Shu*, H. Ding, K.S. Yeo

Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore, Singapore 117576

Received 8 September 2002; revised 20 December 2002; accepted 26 February 2003
Available online 17 April 2004

Abstract

The conventional differential quadrature (DQ) method is limited in its application to regular regions by using functional values along a mesh line to approximate derivatives. In this work, we extend the idea of DQ method to a general case. In other words, any spatial derivative is approximated by a linear weighted sum of all the functional values in the whole physical domain. The weighting coefficients in the new approach are determined by the radial basis functions (RBFs). The proposed method combines the advantages of the conventional DQ method and the RBFs. Since the method directly approximates the derivative, it can be consistently well applied to linear and nonlinear problems. It also remains mesh free feature of RBFs. Numerical examples of linear and nonlinear cases showed that RBF-based DQ method has a potential to become an efficient approach for solving partial differential equations.

Keywords: Differential quadrature method; Radial basis function; Partial differential equation

1. Introduction

The differential quadrature (DQ) method was introduced by Richard Bellman and his associates in the early of 1970s [1,2], following the idea of integral quadrature. 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. The key procedure in the DQ method is the determination of weighting coefficients. As shown by Shu and Richards [3], when the solution of a partial differential equation (PDE) is approximated by a high order polynomial, the weighting coefficients can be computed by a simple algebraic formulation or by a recurrence relationship. Later, Shu and Chew [4] also showed that when the solution of the PDE is approximated by a Fourier series expansion, the weighting coefficients of the first- and second-order derivatives can be computed explicitly by algebraic formulations. The details of the polynomial-based and Fourier series expansion-based DQ methods can be found in the book of Shu [5]. Currently, the DQ method has been extensively applied in engineering for the rapid and accurate solution of various linear and nonlinear differential equations [3–11].

On the other hand, it is noted that the function approximation (polynomial or Fourier series expansion) in the DQ method is along a straight line. This means that numerical discretization of derivatives by the DQ method is also along a straight line. Due to this feature, the DQ method can be directly applied to regular regions, such as rectangular and circular domains. For complex geometry, the DQ method cannot be directly applied. One has to rely on the coordinate transformation technique [9–11]. In this technique, the irregular domain in the physical space is first mapped to a regular domain in the computational space. Then the differential equations and their associated boundary conditions are transformed into relevant forms in the computational space. The numerical discretization is only made in the computational space by the DQ method. Although this technique can obtain very good results for problems with complex geometry, we have to admit that the process is very complicated, and the approach is not as flexible as the finite element method. Practically, there is a demanding to develop a more efficient method for solving complex problems.

It was found that the need of coordinate transformation by the DQ method for complex problems is actually due to its discretization along a straight line. Furthermore, we found that the DQ discretization along a straight line is due...
to one-dimensional function approximation used [5]. It is expected that if a two-dimensional polynomial is used to approximate a function, the DQ approximation of a derivative can involve any point on the two-dimensional plane. And as a consequence, no coordinate transformation is needed. This is the idea of differential cubature (DC) method [12]. Unfortunately, due to oscillatory feature of high order polynomials, the DC method can only obtain stable solution of a PDE by using a limited number of mesh points. It seems that the multi-dimensional polynomial approximation as the test function may not be a good choice in the DQ approximation. As will be shown in this paper, the radial basis functions (RBFs), which have ‘truly’ meshless property and insensitivity to high dimension, could be a good choice in the DQ approximation.

RBFs have been under intensive research as a technique for multivariate data and function interpolation in the past decades, especially in multi-dimensional applications. Their performance demonstrates that RBFs constitute a powerful framework for interpolating or approximating data on nonuniform grids. Furthermore, Buhmann and Micchelli [13] showed that RBFs are attractive for pre-wavelet construction due to their exceptional rates of convergence and infinite differentiability. Since RBFs have excellent performance for function approximation, many researchers turn to explore their ability for solving PDEs. The first trial of such exploration was made by Kansa [14,15]. As shown in Ref. [14], using RBFs as a meshless collocation method to solve PDEs possesses the following advantages: (1) first of all, it is a truly mesh-free method, and is independent of spatial dimension in the sense that the convergence order is of $O(h^{d+1})$, where $h$ is the density of the collocation points and $d$ is the spatial dimension; (2) furthermore, in the context of scattered data interpolation, it is known that some RBFs have spectral convergence. In other words, as the spatial dimension of the problem increases, the convergence order also increases, and hence, much fewer scattered collocation points will be needed to maintain the same accuracy as compared with conventional finite difference, finite element and finite volume methods. This shows the applicability of the RBFs for solving high-dimensional problems. Apart from the work of Kansa [14,15], other researchers, such as Fornberg et al. [16], Hon and Wu [17] also made a great contribution in this area. It should be indicated that although some excellent results were obtained, all previous works related to the application of RBFs for the numerical solution of PDEs are actually based on the function approximation instead of derivative approximation. In other words, these works directly substitute the expression of function approximation by RBFs into a PDE, and then change the dependent variables into the coefficients of function approximation. The process is very complicated, especially for nonlinear problems. For the nonlinear case, some special techniques such as numerical continuation and bifurcation approach proposed by Fedoseyev et al. [18] have to be used to solve the resultant nonlinear equations. Since the techniques are very complicated, it is not easy to apply them for solving practical problems such as fluid dynamics, which usually require a large number of mesh points for accurate solution.

As will be shown in this paper, the advantages of the DQ approximation and RBFs can be combined to provide an efficient discretization method, which is a derivative approximation approach and is mesh-free. In our proposed method, the RBFs are taken as the test functions in the DQ approximation to compute the weighting coefficients. Once the weighting coefficients are computed, the solution process for a PDE by the new method is exactly the same as the conventional DQ method and finite difference schemes. Moreover, the new method can be consistently well applied to linear and nonlinear problems. Our numerical experiments demonstrate that this new method not only inherits the advantages of the DQ method such as high accuracy and efficient computation, but also owns the merits of RBFs such as mesh-free feature and easy extension to high dimension. This article is the first of a series works. We hope to present a new framework in applying the DQ method to practical problems.

2. Development of RBF-based DQ method

The DQ method is a numerical discretization technique for approximation of derivatives. It was initiated from the idea of conventional integral quadrature. The conventional DQ method is applied along a mesh line. As shown by Shu and co-workers [3–5], the conventional DQ method is actually based on the one-dimensional function approximation. The weighting coefficients in the DQ approximation can be computed under the analysis of function approximation and linear vector space. In this work, we extend the idea of DQ approximation to a general case. The limitation on using functional values along a mesh line in the conventional DQ method is released. In our approach, any spatial derivative is approximated by a linear weighted sum of all the functional values in the whole physical domain. The weighting coefficients in our method are determined by RBF approximation and linear vector space analysis. In the following, we will show the details of this approach.

2.1. Derivative approximation by differential quadrature method

It is well known that any integral over a closed domain can be approximated by a linear weighted sum of all the functional values in the integral domain. Following this idea, Bellman et al. [1,2] suggested that the partial derivative of a function with respect to an independent variable can be approximated by a linear weighted sum of functional values at all mesh points in that direction. As shown in Fig. 1, DQ approximates the derivative of a function with respect to $x$ at a mesh point $(x_i, y_j)$
technique must be introduced. To remove this drawback, we need to develop a more efficient approach.

It is noted that the basic idea of the DQ method is that any derivative can be approximated by a linear weighted sum of functional values at some mesh points. We can keep this idea but release the choice of functional values along a mesh line in the conventional DQ approximation. In other words, for a two-dimensional problem shown in Fig. 1, any spatial derivative is approximated by a linear weighted sum of all the functional values in the whole two-dimensional domain. In this approximation, a mesh point in the two-dimensional domain is represented by one index, \( k \), while in the conventional DQ approximation like Eq. (1), the mesh point is represented by two indexes \( i,j \). If the mesh is structured, it is easy to establish the relationship between \( i,j \) and \( k \). For the example shown in Fig. 1, \( k \) can be written as \( k = (i-1)M+j \), \( i = 1,2,\ldots,N; \quad j = 1,2,\ldots,M \). Clearly, when \( i \) is changed from 1 to \( N \) and \( j \) is changed from 1 to \( M \), \( k \) is changed from 1 to \( NM = N \times M \). The new DQ approximation for the \( n \)th order derivative with respect to \( x \), \( f^{(n)}_x \), and the \( m \)th order derivative with respect to \( y \), \( f^{(m)}_y \), at \((x_i,y_j)\) can be written as

\[
\begin{align*}
  f^{(n)}_x(x_i,y_j) &= \sum_{k=1}^{N} w^{(n)}_{i,k} f(x_i,y_j) \\
  f^{(m)}_y(x_i,y_j) &= \sum_{k=1}^{M} w^{(m)}_{j,k} f(x_i,y_j)
\end{align*}
\]  

where \( N,M \) are, respectively, the number of mesh points in the \( x \) and \( y \) direction, \( w^{(n)}_{i,k}, w^{(m)}_{j,k} \) are the DQ weighting coefficients in the \( x \) and \( y \) directions. As shown in Ref. [5], \( w^{(n)}_{i,k} \) depends on the approximation of the one-dimensional function \( f(x,y) \) (\( x \) is the variable), while \( w^{(m)}_{j,k} \) depends on the approximation of the one-dimensional function \( f(x,y) \) (\( y \) is the variable). When \( f(x,y) \) or \( f(x_i,y) \) is approximated by a high order polynomial, Shu and Richards [3] derived a simple algebraic formulation and a recurrence relationship to compute \( w^{(n)}_{i,k} \) and \( w^{(m)}_{j,k} \). When the function is approximated by a Fourier series expansion, Shu and Chew [4] also derived simple algebraic formulations to compute the weighting coefficients of the first- and second-order derivatives. For simple geometry, the above DQ approach can obtain very accurate results by using a considerably small number of mesh points. However, as mentioned in Section 1, for complex geometry, the above scheme cannot be applied directly. The coordinate transformation
it was shown that multiquadrics have the excellent properties for the interpolation of 2D scattered data. Therefore, they are highly recommended as test functions for the numerical schemes proposed in Section 3. Multiquadric RBFs (for generality, it is written with a variable shape parameter) can be written as

$$
\varphi(x, \xi_j) = \varphi(r_j) = \sqrt{r_j^2 + \xi_j^2}
$$

(4)

where $r_j = \|x - \xi_j\|$ is the Euclidian norm. Since $\varphi$ given by Eq. (4) is $C^\infty$ continuous, the multiquadric RBFs can be used directly. If $\Psi_q^d$ denotes the space of $d$-variate polynomials of order not exceeding $q$, and letting the polynomials $P_1, \ldots, P_m$ be the basis of $\Psi_q^d$ in $\mathbb{R}^d$, then the polynomial $\varphi(x)$, in Eq. (3), is usually written in the following form

$$
\varphi(x) = \sum_{i=1}^m \xi_i P_i(x)
$$

(5)

where $m = (q + d)!/(d!(q - 1)!)$. To determine the coefficients $(\lambda_1, \ldots, \lambda_N)$ and $(\xi_1, \ldots, \xi_m)$, extra $m$ equations are required in addition to the $N$ equations resulting from the collocating Eq. (3) at the $N$ points. This is ensured by the $m$ conditions for Eq. (3), viz.

$$
\sum_{j=1}^N \lambda_j P_i(\xi_j) = 0, \quad i = 1, \ldots, m
$$

(6)

### 2.3. Determination of weighting coefficients in RBF-based DQ approximation

In this section, we will use the multiquadric RBFs as test functions to determine the weighting coefficients in the new DQ approximation of derivatives for a two-dimensional problem. However, the method can be easily extended to the three-dimensional case as it is dimension-independent, and other RBFs can also be used as test functions.

Suppose that the solution of a PDE is continuous, which can be approximated by multiquadric RBFs, and only a constant is included in the polynomial term $Y(x)$. So, for a two-dimensional case, Eq. (3) can be reduced to

$$
f(x, y) = \sum_{j=1}^N \lambda_j (\varphi(\|x - \xi_j\|) - \varphi(\|x - \theta\|)) + \lambda_{N+1}
$$

(7)

To make the problem be well-posed, one more equation is required. From Eq. (6), we have

$$
\sum_{j=1}^N \lambda_j = 0 \Rightarrow \lambda_j = -\sum_{j=1, j \neq i}^N \lambda_j
$$

(8)

Substituting Eq. (8) into Eq. (7) gives

$$
f(x, y) = \sum_{j=1, j \neq i}^N \lambda_j (\varphi(\|x - \xi_j\|) - \varphi(\|x - \theta\|)) + \lambda_{N+1}
$$

(9)

The number of unknowns in Eq. (9) is $N$. As no confusion rises, $\lambda_{N+1}$ can be replaced by $\lambda_i$, and Eq. (9) can be written as

$$
f(x, y) = \sum_{j=1, j \neq i}^N \lambda_j (\varphi(\|x - \xi_j\|) - \varphi(\|x - \theta\|)) + \lambda_i
$$

(10)

It is easy to see that $f(x, y)$ in Eq. (10) constitutes $N$-dimensional linear vector space $\mathbf{V}^N$ with respect to the operation of addition and multiplication. From the concept of linear independence, the bases of a vector space can be considered as linearly independent subset that spans the entire space. In the space $\mathbf{V}^N$, one set of base vectors is $1, g_j(x, y) = \varphi(\|x - \xi_j\|) - \varphi(\|x - \theta\|), j = 1, \ldots, N$ but $j \neq i$.

From the property of a linear vector space, if all the base functions satisfy the linear Eq. (2), so does any function in the space $\mathbf{V}^N$ represented by Eq. (10). There is an interesting feature. From Eq. (10), while all the base functions are given, the function $f(x, y)$ is still unknown since the coefficients $\lambda_j$ are unknown. However, when all the base functions satisfy Eq. (2), we can guarantee that $f(x, y)$ also satisfies Eq. (2). In other words, we can guarantee that the solution of a PDE approximated by the RBF satisfies Eq. (2). Thus, when the weighting coefficients of DQ approximation are determined by all the base functions, they can be used to discretize the derivatives in a PDE. That is the essence of the RBF-based DQ method.

Substituting all the base functions $1, g_j(x, y) = \varphi(\|x - \xi_j\|) - \varphi(\|x - \theta\|), j = 1, \ldots, N$, but $j \neq i$ into Eq. (2a), we can obtain

$$
0 = \sum_{k=1}^N w^{(n)}_{i,k} \frac{\partial^n g_j(x_k, y_k)}{\partial x^a},
$$

(11a)

$$
\frac{\partial^n g_j(x_k, y_k)}{\partial y^m} = \sum_{k=1}^N w^{(m)}_{i,k} \frac{\partial^n g_j(x_k, y_k)}{\partial y^m},
$$

(11b)

for the given $i$, equation system (11) has $N$ unknowns with $N$ equations. So, solving this equation system can obtain the weighting coefficients $w^{(m)}_{i,k}$. In a similar manner, when all the base functions are substituted into Eq. (2b), the following equation system can be obtained:

$$
0 = \sum_{k=1}^N \tilde{w}^{(m)}_{i,k} \frac{\partial^n g_j(x_k, y_k)}{\partial y^m},
$$

(12a)

$$
\frac{\partial^n g_j(x_k, y_k)}{\partial x^a} = \sum_{k=1}^N \tilde{w}^{(m)}_{i,k} \frac{\partial^n g_j(x_k, y_k)}{\partial x^a},
$$

(12b)

for the given $i$, equation system (12) is used to compute the weighting coefficients $\tilde{w}^{(m)}_{i,k}$ of $y$-derivatives. For multiquadric RBFs,
$g_j(x,y)$ can be written as

$$g_j(x,y) = \sqrt{(x-x_j)^2 + (y-y_j)^2 + c_j^2}$$

$$- \sqrt{(x-x_i)^2 + (y-y_i)^2 + c_i^2}$$

(13)

From Eq. (13), one can easily obtain the first-order derivatives of $g_j(x,y)$ as

$$\frac{\partial g_j(x,y)}{\partial x} = \frac{x-x_j}{\sqrt{(x-x_j)^2 + (y-y_j)^2 + c_j^2}}$$

$$- \frac{x-x_i}{\sqrt{(x-x_i)^2 + (y-y_i)^2 + c_i^2}}$$

(14a)

$$\frac{\partial g_j(x,y)}{\partial y} = \frac{y-y_j}{\sqrt{(x-x_j)^2 + (y-y_j)^2 + c_j^2}}$$

$$- \frac{y-y_i}{\sqrt{(x-x_i)^2 + (y-y_i)^2 + c_i^2}}$$

(14b)

The second- and higher-order derivatives of $g_j(x,y)$ can also be obtained by differentiating Eq. (13).

One of the most attractive properties in the above method is that the weighting coefficients are only related to the test functions and the position of the collocation points. That character is very appealing when we deal with the nonlinear problems. Since the derivatives are directly discretized, the method can be consistently well applied to linear and nonlinear problems. Another attractive property of RBF-based DQ method is that it is naturally mesh free, i.e. all the information required about the nodes in the domain is nothing but their positions.

2.4. Shape parameter $c$

It is well known that the multiquadrics suffer from a trade-off principle, i.e. one can adjust the parameter $c$ to increase the approximation power of the basis, but pays for this by making the problem more and more ill-conditioned. These effects were initially observed for scattered data interpolation, but we will see that they also occur in the numerical solution of PDEs. Thus, the problem of how to select a good value for the parameter $c$ appears in front of us. Several methods for selecting $c$ for the multiquadric interpolants in the two-dimensional case were suggested in the literature.

Hardy [20] used $c = 0.815d$, where $d = (1/N) \sum_{i=1}^{N} d_i$ and $d_i$ is the distance between the $i$th node and its neighbouring node. Franke [19] replaced $d$ by $D/\sqrt{N}$, where $D$ is the diameter of the minimal circle enclosing all supporting points and suggested to use $c = 1.25(D/\sqrt{N})$. Up to now, optimization of shape parameter and its distribution are still under research. In Section 3, we will show the optimal shape parameter $c$ through the trial-and-error process for the solution of PDEs by RBF-based DQ method. To study whether the optimal shape parameter $c$ is varied with different problems, the optimal $c$ achieved in the linear PDE case is also used in the nonlinear case for the same point distribution.

3. Numerical examples

By using the presently developed RBF-based DQ method, several numerical examples are worked out to investigate the numerical characteristics of the proposed method.

3.1. Linear PDE case

Consider the problem of a two-dimensional Poisson equation in a square domain ($0 \leq x \leq 1, 0 \leq y \leq 1$). The governing equation and boundary condition are defined as

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

(15)

$$0 \leq x \leq 1, \quad 0 \leq y \leq 1$$

Boundary condition:

$$u = 0 \quad \text{on } \partial \Omega$$

(16)

The analytical solution for this problem is $u(x,y) = \sin \pi x \sin \pi y$. Two kinds of mesh point distributions are considered for this problem. One is the uniform point distribution and the other is the random point distribution. A typical plot of uniform and random point distributions is shown in Fig. 2. After numerical discretization of Eq. (15) by the RBF-based DQ method, the resultant algebraic equations are solved by the SOR iteration method. The convergence criteria are chosen as $(|u^{n+1} - u^n|/|u^n| + 10^{-8})_{\text{max}} < 10^{-5}$. When the criteria are satisfied, the convergence results are subsequently obtained.

To study the convergence behavior of the method, 52, 65, 84, 100, 138 and 217 points are used for the random point distribution, while mesh sizes of $7 \times 7$, $8 \times 8$, $9 \times 9$, $10 \times 10$, $12 \times 12$, $15 \times 15$ are used for the uniform point distribution. The numerical results are compared with analytical solution. To observe the convergence, three
where the superscripts \((\cdot)\)' and \((\cdot)''\) denote the first- and second-order derivatives of the dependent variable, respectively.

Table 1 presents the relative errors of \(L_2\), \(H^1\), and \(H^2\) in the interior region and on the boundary obtained for the random point distributions. Table 2 shows the relative errors for the uniform point distributions. The optimal parameter \(c\) shown in Tables 1 and 2 is obtained through the trial-and-error process. It can be seen from the two tables that within the certain number of mesh points, the accuracy of numerical results can be improved by increasing the number of mesh points. However, when the number of mesh points is further increased after a critical value, the accuracy of numerical results is decreased. The reason may be due to the fact that, when the number of mesh points is increased, the condition number of the matrix becomes very large and the matrix tends to be ill-conditioned.

From Tables 1 and 2, we can see that accurate numerical results can be obtained by the RBF-based DQ method using very few mesh points. We also notice that the relative errors for the derivative approximation at the boundary points are relatively larger than those in the interior region. This phenomenon has also been observed by Zhang [21]. So, when a practical problem with derivative boundary
condition is considered, we may need to use a more efficient method to implement the boundary condition.

To clearly show the convergence behavior of the RBF-based DQ method for the random and uniform point distributions, the relative errors are plotted against the mesh spacing \( h \):

For the uniform point distribution, \( h \) is the distance of two nearest mesh points. For the random point distribution, \( h \) is defined as

\[
h = \frac{\text{Length}}{\sqrt{N - 1}}
\]

Using this definition, the numerical results obtained on the uniform point distribution are a little bit more accurate than those on the random point distribution. This can be observed from Figs. 3 and 4, which show the convergence of relative errors on the random and uniform point distributions. It can be seen from these two figures that the convergence trends for the solution of the PDE and approximation of the first- and second-order derivatives are similar. Figs. 3 and 4 also reveal that the convergence trends of random and uniform point distributions are similar.

3.2. Nonlinear PDE case

To further validate the performance of RBF-based DQ method, we will apply it to solve nonlinear PDEs in this section. Like the linear case, the resultant algebraic equations from numerical discretization are solved by the SOR iteration method. We will consider two nonlinear examples. The first example is

\[
\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + u \left( \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} \right) - 2(x + y)u = 4
\]

(21)

The exact solution of Eq. (21) in a square domain is

\[
u(x, y) = x^2 + y^2
\]

(22)

Dirichlet boundary condition for the four edges of the square domain can be obtained from Eq. (22), that is

\[
u(x = 0) = y^2, \quad u(x = 1) = 1 + y^2,
\]

\[
u(y = 0) = x^2, \quad u(y = 1) = 1 + x^2
\]

(23)

Numerical results are obtained by employing the same point distributions used in the linear PDE case. The shape parameters \( c \) and the criteria for the convergence are also chosen the same as those in the linear case. From the numerical

<table>
<thead>
<tr>
<th>Table 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative errors obtained for a nonlinear problem on random point distributions</td>
</tr>
<tr>
<td>Shape parameter, ( c )</td>
</tr>
<tr>
<td>1.272</td>
</tr>
<tr>
<td>1.191</td>
</tr>
<tr>
<td>1.037</td>
</tr>
<tr>
<td>0.629</td>
</tr>
<tr>
<td>0.502</td>
</tr>
<tr>
<td>0.156</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative errors obtained for a nonlinear problem on uniform point distributions</td>
</tr>
<tr>
<td>Shape parameter, ( c )</td>
</tr>
<tr>
<td>1.471</td>
</tr>
<tr>
<td>1.392</td>
</tr>
<tr>
<td>1.151</td>
</tr>
<tr>
<td>1.113</td>
</tr>
<tr>
<td>0.675</td>
</tr>
<tr>
<td>0.311</td>
</tr>
</tbody>
</table>
results shown in Tables 3 and 4, we notice that the optimal shape parameter $c$ achieved in the linear case works very well in the nonlinear case. It can also provide very accurate results. From this study, we may expect that, for a fixed mesh point distribution, the optimal shape parameter $c$ in the RBF-based DQ method remains the same for various problems. The convergence trends of this nonlinear case on the random and uniform point distributions are shown in Figs. 5 and 6. As compared with Figs. 3 and 4, we can find that the convergence trend of the nonlinear case is similar to the linear case.

The second example is chosen from Ref. [22]. The governing equation and boundary condition of this problem are

\[-\varepsilon^2 \nabla^2 u - u + u^3 = f, \quad \text{in } \Omega = [0, 1] \times [0, 1] \quad (24)\]

\[u = 0, \quad \text{on } \partial \Omega \quad (25)\]

The source function $f$ on the right-hand side is chosen so that Eq. (24) has an analytic solution of the form

\[u(x, y) = \psi(x) \psi(y) \quad (26)\]

with

\[\psi(t) = 1 + e^{-1/t} - e^{-t} - e^{(t-1)/t} \quad (27)\]

Here, the parameter $\varepsilon$ determines the size of the boundary layers near the edges of the domain $\Omega$. We use a value of $\varepsilon = 0.1$ in our numerical experiments. In this case, we solve the PDE on the random point distribution. The optimal shape parameter $c$ is also taken the same as that we achieved in the linear PDE case. The $L_2$ relative error of this problem is shown in Table 5. From the relative error in the $L_2$ norm, we can see that the RBF-based DQ method also works well for this nonlinear problem. To visualize the numerical solution on the random point distribution, we need to interpolate the results on a uniform grid.

**Table 5**

<table>
<thead>
<tr>
<th>Shape parameter $c$</th>
<th>Total node number</th>
<th>$L_2$ error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.272</td>
<td>52</td>
<td>$1.4474 \times 10^{-2}$</td>
</tr>
<tr>
<td>1.191</td>
<td>65</td>
<td>$8.4122 \times 10^{-3}$</td>
</tr>
<tr>
<td>1.037</td>
<td>84</td>
<td>$7.3988 \times 10^{-3}$</td>
</tr>
<tr>
<td>0.629</td>
<td>100</td>
<td>$1.1071 \times 10^{-2}$</td>
</tr>
<tr>
<td>0.502</td>
<td>138</td>
<td>$9.8744 \times 10^{-3}$</td>
</tr>
<tr>
<td>0.156</td>
<td>217</td>
<td>$9.4433 \times 10^{-3}$</td>
</tr>
</tbody>
</table>
of $101 \times 101$. The interpolation scheme introduced in Eq. (10) is used to do this job. The initial guess, numerical and exact solutions of the problem are shown, respectively, in Figs. 7–9. As compared with the exact solution (shown in Fig. 9), the numerical solution shows no visual difference.

In general, we can observe that for the RBF-based DQ method, even with a small number of mesh points, as shown in Tables 1–4, good accuracy of numerical results can be achieved for both the linear and nonlinear problems.

4. Conclusions

A mesh free radial basis function-based differential quadrature (RBF-based DQ) method is presented in this paper. The proposed method is similar to finite difference schemes in the sense of derivative approximation. It is emphasized that the method permits the preservation of the simplicity of the finite differences, while eliminating their drawbacks when it applies to arbitrary nodal distribution. Furthermore, it also makes no distinction regarding the dimension of the problem. Numerical results showed that our RBF-based DQ scheme is an efficient approach for solution of PDEs.

On the other hand, it should be indicated that when the number of nodes is increased to a certain value, the accuracy of RBF-DQ approximation for derivative will decrease with the increase of nodes. This is because the matrix for computing the weighting coefficients in the DQ approximation would be highly ill-conditioned when the number of nodes is large. To remove this difficulty, a local RBF-DQ approximation for derivative is recommended, which will be discussed in detail in subsequent papers.

References


