INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2007; **53**:969–984
Published online 2 August 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1315

Integrated radial basis functions-based differential quadrature method and its performance

C. Shu*,† and Y. L. Wu

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

SUMMARY

In this paper, indirect radial basis function networks (IRBFN) proposed by Nam and Tranh (*Neural Networks* 2001; **14**(2):185–199; *Appl. Math. Modelling* 2003; **27**:197–220) are incorporated into the differential quadrature (DQ) approximation of derivatives. For simplicity, this new variant of RBF-DQ approach is named as iRBF-DQ method. The proposed approach is validated by its application to solve the one-dimensional Burger's equation, and simulate natural convection in a concentric annulus by solving Navier–Stokes equations. It was found that as compared to the benchmark data, the iRBF-DQ approach can provide more accurate results than the original RBF-DQ method. Copyright © 2006 John Wiley & Sons, Ltd.

Received 18 January 2006; Revised 24 May 2006; Accepted 10 June 2006

KEY WORDS: iRBF-DQ; RBF-DQ; integrated RBF; approximation of derivatives; Burger's equation; natural convection; concentric annuli

1. INTRODUCTION

There are many numerical methods for solving partial differential equations (PDEs) in engineering and science. The conventional methods include finite difference (FD) method, finite element (FE) method, and finite volume (FV) method. These methods usually use the low-order polynomial to approximate the derivatives in PDEs. Therefore, they are classified as low-order methods. To achieve acceptable accuracy, the low-order methods have to use a large number of grid points. On the contrary, their high order counterparts, such as the differential quadrature (DQ) method [1, 2], use just a few grid points to obtain accurate numerical results. However, the high-order

*Correspondence to: C. Shu, Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 117576, Singapore.
†E-mail: mpeshuc@nus.edu.sg

polynomial-based methods may encounter the polynomial oscillation problem, which makes the computation oscillatory, sometimes even diverged.

On the other hand, it was found that radial basis functions (RBFs) are a powerful tool for function approximation [3], especially for interpolation of scattered data points. Due to favourable properties of RBF approximation such as high accuracy and meshless feature, many researchers have made effort to apply it to solve PDEs. The first trial of such exploration was made by Kansa in 1990 [4, 5]. He solved some hyperbolic, parabolic, and elliptic PDEs using multiquadric (MQ) RBFs, and found that RBFs could yield a very accurate solution for parabolic and elliptic PDEs. After that, the RBFs attract more and more attentions on solving PDEs as a global method, and a series of papers were published in the literature [6–12]. Note that the Kansa's RBF method is actually based on the function approximation. To approximate derivatives by using RBFs, Shu and his co-workers [13–15] proposed the RBF-DQ method, which combines the DQ approximation of derivatives and function approximation of RBF. Previous applications showed that RBF-DQ is an efficient method to solve linear and nonlinear PDEs.

Recently, Nam and Tranh [16, 17] proposed the indirect radial basis function networks (IRBFN). They found that the IRBFN performs better than the conventional RBF method. In the conventional RBF method, the RBF approximation is directly applied to the solution function. As such, any derivative of the solution function can be obtained by differentiating the RBF expression. In the IRBFN approach, the RBF approximation is applied to a targeted derivative (first order or second order) of the solution function. Consequently, the solution function is obtained by integrating the derivative with the RBF expression. Obviously, as compared to the conventional RBF method, the procedure of IRBFN is equivalent to use a new set of RBFs which are derived from integration of original RBFs. The new set of RBFs may have better approximation for the targeted derivative. This has been demonstrated by Nam and Tranh [16, 17].

It is interesting to investigate the performance of RBF-DQ method for the solution of PDEs when the solution function is approximated by integrated radial basis functions (iRBF). This motivates the present paper. In this paper, the RBF-DQ method with integrated radial basis functions, iRBF-DQ for short, is presented and validated by its application to solve one-dimensional Burger's equation and two-dimensional Navier–Stokes equations. Numerical results do show that iRBF-DQ method has a better accuracy than its original version of RBF-DQ for solution of PDEs.

2. INTEGRATED RADIAL BASIS FUNCTION-BASED DIFFERENTIAL QUADRATURE (iRBF-DQ) METHOD

iRBF-DQ method is developed from the radial basis function-based differential quadrature (RBF-DQ) method. In the following, we will firstly give a brief description on the RBF-DQ method.

2.1. RBF-DQ method

According to RBF-DQ method [13–15], the derivatives of a smooth function $u(x)$ are approximated by a weighted linear sum of the functional values at neighbouring mesh nodes. For example, the RBF-DQ approximation for the nth order derivative of $u(x)$ with respect to x, $u_x^{(n)}$, at the ith node x_i can be written as

$$u_x^{(n)}(x_i) = \sum_{j=1}^{N} w_{i,j}^{(n)} u(x_j) \tag{1}$$

DOI: 10.1002/fld

where N is the number of nodes used in the supporting region, $w_{i,j}^{(n)}$ are the DQ weighting coefficients. The determination of weighting coefficients is based on the analysis of function approximation and the analysis of a linear vector space.

In the usual procedure of RBF-DQ method, it is supposed that there are N nodes, $x_1, \ldots, x_N$, in the support domain of point x. Then the approximation of u by RBFs can be written as

$$u(x) = \sum_{k=1}^{N} \lambda_k \varphi_k(x) \tag{2}$$

where $\varphi_k(x)$ is a RBF, and λ_k is the coefficient for $\varphi_k(x)$. The three typical forms of $\varphi_k(x)$ are given as below

$$\varphi_k(x) = (r_k^2 + c^2)^{1/2} \quad \text{(multiquadric, MQ)}$$

$$\varphi_k(x) = (r_k^2 + c^2)^{-1/2} \quad \text{(inverse multiquadric, IMQ)}$$

$$\varphi_k(x) = e^{-cr^2} \quad \text{(Gaussian, GS)}$$

where the positive constants c are called shape parameters, r the distance between point x and x_k, i.e.

$$r = |x - x_k| \tag{3}$$

Equation (2) has N freedoms, which can be considered to construct an N-dimensional linear vector space V_N. Obviously, the RBFs $\{\varphi_k(x),\ k = 1, \ldots, N\}$ in Equation (2) is a set of base functions in V_N, which can be taken as the base vectors. Therefore, the weighting coefficients $w_{i,j}^{(n)}$ can be determined using the same way as in the conventional DQ method. Substituting all the base functions into Equation (1) leads to

$$\frac{\partial^{(n)} \varphi_k(x_i)}{\partial x^{(n)}} = \sum_{j=1}^{N} w_{i,j}^{(n)} \varphi_k(x_j) \tag{4}$$

The above equation can be further put in the matrix form,

$$\underbrace{\begin{bmatrix} \dfrac{\partial^{(n)} \varphi_1(x_i)}{\partial x^{(n)}} \\ \dfrac{\partial^{(n)} \varphi_2(x_i)}{\partial x^{(n)}} \\ \vdots \\ \dfrac{\partial^{(n)} \varphi_N(x_i)}{\partial x^{(n)}} \end{bmatrix}}_{\frac{\partial \vec{\varphi}(\mathbf{x}_i)}{\partial x}} = \underbrace{\begin{bmatrix} \varphi_1(x_1) & \varphi_1(x_2) & \cdots & \varphi_1(x_N) \\ \varphi_2(x_1) & \varphi_2(x_2) & \cdots & \varphi_2(x_N) \\ & & \ddots & \\ \varphi_N(x_1) & \varphi_2(x_2) & \cdots & \varphi_N(x_N) \end{bmatrix}}_{[A]} \underbrace{\begin{bmatrix} w_{i1}^{(n)} \\ w_{i2}^{(n)} \\ \vdots \\ w_{iN}^{(n)} \end{bmatrix}}_{\vec{w}_i} \tag{5}$$

DOI: 10.1002/fld

According to the theory of RBF approximation, we know that the matrix [A] could be invertible if the appropriate φ is chosen. So, the vector $\vec{w}_i^{(n)}$ can be obtained by,

$$\vec{w}_i^{(n)} = [\mathrm{A}]^{-1} \frac{\partial^{(n)} \vec{\varphi}(x_i)}{\partial x^{(n)}}, \quad i = 1, \ldots, N \tag{6}$$

2.2. iRBF-DQ method with polynomial term

According to the work of Nam and Tranh [16, 17], if the second-order derivative $u''(x)$ is approximated by the original RBFs, i.e.

$$u''(x) = \sum_{k=1}^{N} \mu_k \varphi_k(x) \tag{7}$$

then the first-order derivative $u'(x)$ can be obtained by integration, which is written as

$$u'(x) = \int u''(x)\,\mathrm{d}x = \sum_{k=1}^{N} \mu_k H_k(x) + C_2 \tag{8}$$

Similarly, the function u is obtained as

$$u(x) = \int u'(x)\,\mathrm{d}x = \sum_{k=1}^{N} \mu_k \overline{H}_k(x) + C_2 x + C_1 \tag{9}$$

Here $H_k(x)$ and $\overline{H}_k(x)$ are the iRBFs, and μ_k is the coefficient for $H_k(x)$ and $\overline{H}_k(x)$, C_1 and C_2 are the constants of integration.

It is found that the multiquadric radial basis function (MQ-RBF) is the best performer in approximation of function and its derivatives. Therefore, in this paper, we only consider the MQ-RBF. The expression for the integrated MQ-RBF, according to Nam and Tranh [16, 17], can be written as:

$$H(x) = \int \varphi(x)\,\mathrm{d}x = \frac{x\sqrt{x^2 + c^2}}{2} + \frac{c^2}{2} \ln(x + \sqrt{x^2 + c^2}) \tag{10}$$

$$\overline{H}(x) = \int H(x)\,\mathrm{d}x = \frac{(x^2 + c^2)^{3/2}}{6} + \frac{c^2 x}{2} \ln(x + \sqrt{x^2 + c^2}) - \frac{c^2}{2}\sqrt{x^2 + c^2} \tag{11}$$

It is clear that the approximation by Equation (9) is actually the following form:

$$u(x) = \sum_{k=1}^{N} \mu_k \overline{H}_k(x) + P(x) \tag{12}$$

where $P(x)$ is the polynomial with $P(x) = C_2 x + C_1$.

In the work of Nam and Tranh [16, 17], the weights μ_k and the constants of integration C_1 and C_2 are determined by minimizing the sum of squared errors which is defined as

$$J = \text{sum of squared errors} = \sum_{i=1}^{m} \left[u(x_i) - \sum_{k=1}^{N} \mu_k \overline{H}_k(x_i) - C_2 x_i - C_1 \right]^2 \tag{13}$$

DOI: 10.1002/fld

where m is the number of the sample points, $u(x_i)$ is the function value at x_i. By setting

$$\frac{\partial J}{\partial \mu_k}=0, \quad k=1,\ldots,m, \quad \frac{\partial J}{\partial C_1}=0, \quad \frac{\partial J}{\partial C_2}=0$$

we can get an algebraic equation system for μ_k and C_1, C_2. Nam and Tranh used the singular value decomposition (SVD) method to solve this equation system. After getting the solution for the coefficients μ_k and constants C_1, C_2, the second and first-order derivatives as well as the function can be obtained by Equations (7)–(9).

Under the frame of RBF-DQ method, the approximation of derivatives is given by Equation (1). The only task here is to find a set of basis functions to determine the DQ weighting coefficients $w_{i,j}^{(n)}$. In the present iRBF-DQ method, the second-order integrated MQ-RBFs $\overline{H}_k(x)$ are used as base functions. Unlike the IRBFN method [16, 17] in which the coefficients are determined by SVD, in our method, the coefficients in Equation (12) are determined directly by collocation at N nodes within the support domain. From Equation (12), it is found that there are $N+2$ unknown coefficients to be determined but only N equations are available. Therefore, it is not a well-posed problem. To make the equation system be a closed one, the following constraints are enforced:

$$\sum_{k=1}^{N} \mu_k = 0 \tag{14}$$

$$\sum_{k=1}^{N} \mu_k x_k = 0 \tag{15}$$

As a result, we can get

$$\mu_1 = -\sum_{k=2}^{N} \mu_k \tag{16}$$

$$\sum_{k=1}^{N} \mu_k x_k = 0 \Rightarrow \mu_1 x_1 + \sum_{k=2}^{N} \mu_k x_k = 0 \Rightarrow \sum_{k=2}^{N} \mu_k (x_k - x_1) = 0$$

$$\Rightarrow \mu_2 (x_2 - x_1) + \sum_{k=3}^{N} \mu_k (x_k - x_1) = 0 \Rightarrow \mu_2 = -\sum_{k=3}^{N} \mu_k \frac{x_k - x_1}{x_2 - x_1} \tag{17}$$

Substituting Equation (16) into Equation (12) gives

$$\begin{aligned} u(x) &= \mu_1 \overline{H}_1(x) + \sum_{k=2}^{N} \mu_k \overline{H}_k(x) + P(x) \\ &= \sum_{k=2}^{N} \mu_k [\overline{H}_k(x) - \overline{H}_1(x)] + P(x) \\ &= \mu_2 [\overline{H}_2(x) - \overline{H}_1(x)] + \sum_{k=3}^{N} \mu_k [\overline{H}_k(x) - \overline{H}_1(x)] + P(x) \end{aligned} \tag{18}$$

Substituting Equation (17) into Equation (18) gives

$$u(x) = \sum_{k=3}^{N} \mu_k \left\{ [\overline{H}_k(x) - \overline{H}_1(x)] - \frac{x_k - x_1}{x_2 - x_1} [\overline{H}_2(x) - \overline{H}_1(x)] \right\} + C_2 x + C_1 \tag{19}$$

DOI: 10.1002/fld

The number of unknowns in Equation (12) is now reduced to N, because constants μ_1 and μ_2 have been removed by using Equations (16) and (17). As no confusion rises, C_1 and C_2 can be replaced by μ_1 and μ_2, and Equation (19) can be re-written as

$$u(x)=\sum_{k=3}^{N}\mu_k\left\{[\overline{H}_k(x)-\overline{H}_1(x)]-\frac{x_k-x_1}{x_2-x_1}[\overline{H}_2(x)-\overline{H}_1(x)]\right\}+\mu_2 x+\mu_1 \tag{20}$$

By setting

$$g_k(x)=\left\{[\overline{H}_k(x)-\overline{H}_1(x)]-\frac{x_k-x_1}{x_2-x_1}[\overline{H}_2(x)-\overline{H}_1(x)]\right\} \tag{21}$$

Equation (20) can be further written as

$$u(x)=\mu_1+\mu_2 x+\sum_{k=3}^{N}\mu_k g_k(x) \tag{22}$$

The form of Equation (22) constructs an N-dimensional linear vector space V_N. A set of base functions in V_N can be taken as

$$q_1=1,\quad q_2=x,\quad q_k(x)=g_k(x),\quad k=3,\ldots,N \tag{23}$$

Using the same procedure as in the original RBF-DQ method (described in Section 2.1), the weighting coefficients of the nth order derivative of $u(x)$ can be determined. Take the first-order derivative as an example. The matrix form of the weighting coefficients can be written as

$$\vec{q}_i^{(1)}=[G]\vec{w}_i^{(1)} \tag{24}$$

where

$$\vec{q}_i^{(1)}=\begin{bmatrix}0\\1\\g_3^{(1)}(x_i)\\\vdots\\g_N^{(1)}(x_i)\end{bmatrix},\quad [G]=\begin{bmatrix}1&1&\cdots&1\\x_1&x_2&\cdots&x_N\\g_3(x_1)&g_3(x_2)&\ddots&g_3(x_N)\\\vdots&\vdots&\cdots&\vdots\\g_N(x_1)&g_N(x_2)&\ldots&g_N(x_N)\end{bmatrix},\quad \vec{w}_i^{(1)}=\begin{bmatrix}w_{i1}^{(1)}\\w_{i2}^{(1)}\\w_{i3}^{(1)}\\\vdots\\w_{iN}^{(1)}\end{bmatrix}$$

Then the weighting coefficients for the first-order derivative can be obtained by the following equation:

$$\vec{w}_i^{(1)}=[\mathbf{G}]^{-1}\vec{q}_i^{(1)} \tag{25}$$

The weighting coefficients for the nth order derivative $w_{i,j}^{(n)}$ can be obtained in the same way.

2.3. iRBF-DQ method without polynomial term

One of the advantages of RBF-DQ method is that, we can calculate the DQ weighting coefficients $w_{i,j}^{(n)}$ as long as we can find N base functions in V_N, and construct a brand new RBF-DQ variant.

DOI: 10.1002/fld

In the current iRBF-DQ method, the approximation of u by iRBF $\overline{H}_k(x)$ without polynomial term can be written as

$$u(x) = \sum_{k=1}^{N} \mu_k \overline{H}_k(x) \tag{26}$$

By comparing Equation (12) with Equation (26), it is found that the polynomial term $P(x)$ appended in Equation (12) ($P(x) = C_2 x + C_1$) is deleted. This brings a great convenience into the computation.

It is found that Equation (26) has N freedoms, which can be considered to construct an N-dimensional linear vector space V_N. Obviously, the iRBFs $\{\overline{H}_k(x), k = 1, \ldots, N\}$ in Equation (26) form a set of base functions in V_N. Therefore, the weighting coefficients $w_{i,j}^{(n)}$ can be determined using the same way as in the previous sections, and can be written as

$$\vec{w}_i^{(n)} = [B]^{-1} \frac{\partial^{(n)} \vec{\overline{H}}(x_i)}{\partial x^{(n)}}, \quad i = 1, \ldots, N \tag{27}$$

where

$$[B] = \begin{bmatrix} \overline{H}_1(x_1) & \overline{H}_1(x_2) & \cdots & \overline{H}_1(x_N) \\ \overline{H}_2(x_1) & \overline{H}_2(x_2) & \cdots & \overline{H}_2(x_N) \\ & & \ddots & \\ \overline{H}_N(x_1) & \overline{H}_2(x_2) & \cdots & \overline{H}_N(x_N) \end{bmatrix}$$

3. APPLICATION TO MODEL PROBLEMS

With the derivative approximation, the iRBF-DQ method can be used to solve the PDEs. The proposed two versions of iRBF-DQ method are first validated by solving the one-dimensional Burger's equation, which is written as

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \varepsilon \frac{\partial^2 u}{\partial x^2}, \quad x \in [0, 1], \quad t \in [0, T] \tag{28}$$

with initial condition

$$u(x, 0) = f(x) \tag{29}$$

where ε is a constant, T is a specified time. To obtain the analytical solution of Equation (28) for comparison purpose, by using the following transformation:

$$u(x, t) = -2 \cdot \varepsilon \cdot \frac{\partial w(x, t)}{\partial x} \Big/ w(x, t) \tag{30}$$

$$f(x) = -2 \cdot \varepsilon \cdot \frac{\mathrm{d}g(x)}{\mathrm{d}x} \Big/ g(x) \tag{31}$$

DOI: 10.1002/fld

Equation (28) can be reduced to a linear heat conduction equation as follows:

$$\frac{\partial w}{\partial t} = \varepsilon \frac{\partial^2 w}{\partial x^2} \tag{32}$$

with $w(x, 0) = g(x)$.

For the test case here, $f(x)$ is chosen as

$$f(x) = -2\varepsilon \frac{v_1 \pi \cos(\pi x) + \frac{1}{2} v_2 \pi \cos(\frac{1}{2}\pi x)}{v_1 \sin(\pi x) + v_2 \sin(\frac{1}{2}\pi x) + v_3} \tag{33}$$

Then the analytical solution of Equation (32) can be expressed as

$$w(x, t) = v_1 \cdot \exp(-\varepsilon \pi^2 t) \cdot \sin(\pi x) + v_2 \cdot \exp(-0.25\varepsilon \pi^2 t) \cdot \sin(0.5\pi x) + v_3 \tag{34}$$

where v_1, v_2, v_3 are the constants and chosen as $v_1 = 0.2$, $v_2 = 0.1$, $v_3 = 0.3$, $\varepsilon = 0.01$ in the present work.

The two versions of the iRBF-DQ formulation, that is, iRBF-DQ method with and without polynomial term, are used to discretize Equation (28). The resulting ordinary differential equation is solved by the fourth-order Runge–Kutta scheme.

It is well known that the accuracy of RBF approximation depends heavily on the choice of the shape parameter c. In our work, we take the expression of c as follows:

$$c = c0 \cdot \Delta x_i \tag{35}$$

where Δx_i is the shortest distance between the node i and its neighbouring nodes. To some extent, Δx_i represents the information of the nodal distribution in the domain. $c0$ is a dimensionless shape parameter, which is irrelevant of Δx_i. In this paper, different node distributions are considered to determine the valid range of the shape parameter $c0$ for iRBF-DQ with and without polynomial term. The original RBF-DQ method proposed in References [13–15] is also applied to solve the same problem for the purpose of comparison. In present computation, two kinds of node distributions are adopted. They are:

(1) Type 1: 17 points are distributed on the line uniformly;
(2) Type 2: 17 points are distributed according to Reference [18], i.e.

$$x_i = \frac{1}{2}\left(1 - \cos\frac{i-1}{N-1}\pi\right) L, \quad i = 1, 2, \ldots, N \tag{36}$$

where L is the length of the computational domain $0 \leqslant x \leqslant L$. Here $L = 1$.

Figures 1 and 2 show the curves of the average relative errors *err* against different $c0$ for the three RBF-DQ methods with above two kinds of node distributions. It was found that for all the three methods, a very small or a very large $c0$ would lead a large numerical error. It was also found, in this computation, that when $c0$ is larger than a critical value, such as when $c0 > 13$ for all the three methods with uniformly distributed points, the numerical errors increase. The reason may be due to the ill-condition of resultant equation system for the weighting coefficients.

From the observation of Figures 1, 2, it is found that the numerical results of present iRBF-DQ methods are more accurate than those of the original RBF-DQ method. According to Nam and

DOI: 10.1002/fld

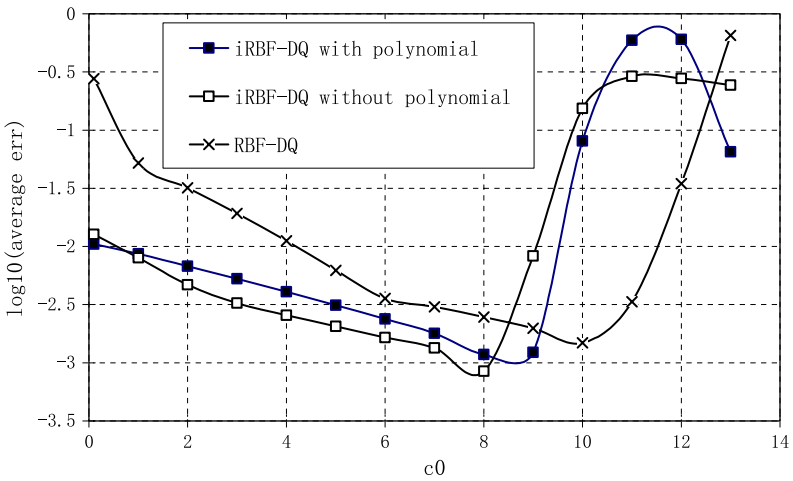

Figure 1. Average relative error vs $c0$ for solution of Burger equation with use of uniform grid.

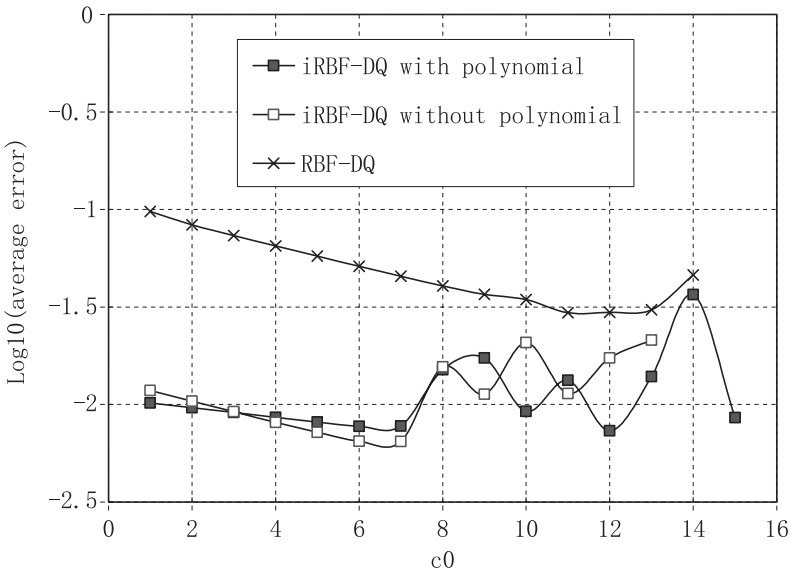

Figure 2. Average relative error vs $c0$ for solution of Burger equation with use of non-uniform grid.

Tranh [16, 17], this may be due to the fact that the process of integration of RBF has the property of damping out or at least containing the inherent inaccuracy, while the process of differentiation may magnify the inaccuracy.

It is also found that for the two versions of iRBF-DQ method, the accuracy of the solution is at the same level. For some cases, such as uniform mesh, the performance of the iRBF-DQ method

DOI: 10.1002/fld

Table I. Unsteady solution of Burger's equation (uniform mesh, $c0$ = optimal value).

		Computational error			
t	x	iRBF-DQ with polynomial ($c0=9$)	iRBF-DQ without polynomial ($c0=8$)	RBF-DQ ($c0=10$)	Analytical
0.1	0.0	−0.051924	−0.051922	−0.051912	−0.051923
	0.5	−0.003897	−0.003897	−0.003897	−0.003897
	1.0	0.031128	0.031130	0.031133	0.031127
0.5	0.0	−0.050289	−0.050225	−0.050118	−0.050215
	0.5	−0.003917	−0.003917	−0.003917	−0.003917
	1.0	0.029984	0.030034	0.030063	0.029995
1.0	0.0	−0.048473	−0.048244	−0.047946	−0.048168
	0.5	−0.003939	−0.003939	−0.003939	−0.003939
	1.0	0.028561	0.028761	0.028854	0.028638

Table II. Unsteady solution of Burger's equation (nonuniform mesh, $c0$ = optimal value).

		Computational error			
t	x	iRBF-DQ with polynomial ($c0=6$)	iRBF-DQ without polynomial ($c0=5$)	RBF-DQ ($c0=11$)	Analytical
0.1	0.0	−0.051833	−0.051917	−0.051467	−0.051923
	0.5	−0.003898	−0.003898	−0.003896	−0.003897
	1.0	0.031168	0.031196	0.030903	0.031127
0.5	0.0	−0.049657	−0.050196	−0.047826	−0.050215
	0.5	−0.003917	−0.003917	−0.003920	−0.003917
	1.0	0.030312	0.030536	0.028768	0.029995
1.0	0.0	−0.047188	−0.048205	−0.044202	−0.048168
	0.5	−0.003939	−0.003939	−0.003952	−0.003939
	1.0	0.029337	0.029878	0.026513	0.028638

without polynomial term is even better than the iRBF-DQ with polynomial term. This implies that for the iRBF-DQ method, the constants of integration in Equations (8) and (9) may not be necessary although they cannot be avoided in the IRBFN method. It seems that RBF-DQ method has more flexibility than other RBF-related schemes to approximate the derivatives.

Tables I and II list the computational results obtained by the two iRBF-DQ methods as well as the original RBF-DQ method with their optimal values of $c0$. The corresponding analytical results are also included in the table for comparison. The time step size was chosen as 0.001. Clearly, the numerical solutions of the iRBF-DQ method are very accurate. In the next section, the iRBF-DQ method will be applied to solve more complex flow problems.

DOI: 10.1002/fld

4. SIMULATION OF NATURAL CONVECTION IN CONCENTRIC ANNULI BY iRBF-DQ METHOD

In this section, we will apply the iRBF-DQ method to simulate the natural convection in an annulus between two concentric circular cylinders.

In the present work, we only apply the iRBF-DQ method in the radial direction. The node distribution along the radial direction is taken the same as shown in Equation (36) so that we can use the optimal value of $c0$ obtained from last section. The uniform mesh is taken along the circumferential direction and the derivatives in the θ direction are discretized by the second-order finite difference scheme. The two iRBF-DQ versions introduced previously, that is, iRBF-DQ with polynomial term and iRBF-DQ without polynomial term, are used to discretize all the derivatives in the r direction. According to the test study for solution of Burger's equation in this work, the dimensionless shape parameter $c0$ for iRBF-DQ with polynomial term and iRBF-DQ without polynomial term are taken as 5.0 and 5.0, respectively.

4.1. Governing equations and numerical discretization

The governing equations in terms of the vorticity-stream function formulation can be written in the cylindrical coordinate system as

$$\frac{\partial^2\psi}{\partial r^2}+\frac{1}{r}\frac{\partial\psi}{\partial r}+\frac{1}{r^2}\frac{\partial^2\psi}{\partial\theta^2}=\omega \tag{37}$$

$$\frac{\partial\omega}{\partial t}+u\frac{\partial\omega}{\partial r}+\frac{v}{r}\frac{\partial\omega}{\partial\theta}=Pr\left(\frac{\partial^2\omega}{\partial r^2}+\frac{1}{r}\frac{\partial\omega}{\partial r}+\frac{1}{r^2}\frac{\partial^2\omega}{\partial\theta^2}\right)-Pr\cdot Ra\left(\sin\theta\frac{\partial T}{\partial r}+\frac{1}{r}\cos\theta\frac{\partial T}{\partial\theta}\right) \tag{38}$$

$$\frac{\partial T}{\partial t}+u\frac{\partial T}{\partial r}+\frac{v}{r}\frac{\partial T}{\partial\theta}=\frac{\partial^2 T}{\partial r^2}+\frac{1}{r}\frac{\partial T}{\partial r}+\frac{1}{r^2}\frac{\partial^2 T}{\partial\theta^2} \tag{39}$$

where $u=\frac{1}{r}\frac{\partial\psi}{\partial\theta}$, $v=-\frac{\partial\psi}{\partial r}$

The dimensionless parameters appeared in the above equations are the Prandtl number, $Pr=\mu C_p/k$, the Rayleigh number, $Ra=C_p\rho_0 g\beta L^3(T_\mathrm{i}-T_\mathrm{o})/kv$, where T_i and T_o are the temperature on the inner and outer cylinder, respectively. The inner cylinder is assumed to be heated. The length of the cylinders is assumed to be infinite, thus the flow and heat transfer in the annulus are regarded as two dimensional. The boundary conditions on two impermeable isothermal walls are given by

$$\psi=u=v=0,\quad \omega=\frac{\partial^2\psi}{\partial r^2},\quad T=1 \tag{40}$$

on the inner cylinder and

$$\psi=u=v=0,\quad \omega=\frac{\partial^2\psi}{\partial r^2},\quad T=0 \tag{41}$$

on the outer cylinder. The periodic condition is implemented in the θ direction, which can be written as

$$f(r,2\pi)=f(r,0)$$

where $f=(\psi,u,v,\omega,T)$

DOI: 10.1002/fld

As mentioned earlier, in this work, the iRBF-DQ method is only applied in the r direction. In the θ direction, the derivatives are discretized by the conventional second-order finite difference scheme. As a result, Equations (37)–(39) can be discretized at a mesh point (θ_i, r_j) as

$$\sum_{k=1}^{M} b_{j,k}\psi_{i,k} + \frac{1}{r_j}\sum_{k=1}^{M} a_{j,k}\psi_{i,k} + \frac{\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}}{r_j^2 \Delta\theta^2} = \omega_{i,j} \tag{42}$$

$$\begin{aligned}\frac{\mathrm{d}\omega_{i,j}}{\mathrm{d}t} + u_{i,j}\sum_{k=1}^{M} a_{j,k}\omega_{i,k} + \frac{v_{i,j}}{r_j}\frac{\omega_{i+1,j} - \omega_{i-1,j}}{2\Delta\theta} \\ = Pr \cdot \left[\sum_{k=1}^{M} b_{j,k}\omega_{i,k} + \frac{1}{r_j}\sum_{k=1}^{M} a_{j,k}\omega_{i,k} + \frac{\omega_{i+1,j} - 2\omega_{i,j} + \omega_{i-1,j}}{r_j^2 \Delta\theta^2}\right] \\ -Pr \cdot Ra\left[\sin\theta_i \sum_{k=1}^{M} a_{j,k}T_{i,k} + \frac{1}{r_j}\cos\theta_i \frac{T_{i+1,j} - T_{i-1,j}}{2\Delta\theta}\right]\end{aligned} \tag{43}$$

$$\begin{aligned}\frac{\mathrm{d}T_{i,j}}{\mathrm{d}t} + u_{i,j}\sum_{k=1}^{M} a_{j,k}T_{i,k} + \frac{v_{i,j}}{r_j}\frac{T_{i+1,j} - T_{i-1,j}}{2\Delta\theta} \\ = \sum_{k=1}^{M} b_{j,k}T_{i,k} + \frac{1}{r_j}\sum_{k=1}^{M} a_{j,k}T_{i,k} + \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{r_j^2 \Delta\theta^2}\end{aligned} \tag{44}$$

where M is the number of mesh points in the r direction, $a_{j,k}$ and $b_{j,k}$ are the iRBF-DQ weighting coefficients of the first and second-order derivatives with respect to r, which can be computed as shown in the previous section.

In a similar manner, the derivatives in the boundary condition equations (40) and (41) can be discretized by the iRBF-DQ method. The time derivatives in Equations (42)–(44) can be discretized by Euler implicit scheme. The resultant algebraic equations are solved by SOR method.

4.2. Numerical results and discussion

Both iRBF-DQ methods are applied to do the simulation. The computed values of average equivalent conductivities are used to compare the present results with available data in the literature. The average equivalent conductivity is defined as [19]

$$\overline{k}_{\mathrm{eqi}} = -\frac{\ln(rr)}{2\pi(rr-1)}\int_0^{2\pi} \frac{\partial T}{\partial r}\cdot \mathrm{d}\theta \tag{45}$$

for the inner cylinder, and

$$\overline{k}_{\mathrm{eqo}} = -\frac{rr\cdot\ln(rr)}{2\pi(rr-1)}\int_0^{2\pi} \frac{\partial T}{\partial r}\cdot \mathrm{d}\theta \tag{46}$$

for the outer cylinder. Table III compares the computed $\overline{k}_{\mathrm{eqi}}$ and $\overline{k}_{\mathrm{eqo}}$ by iRBF-DQ method for the case of $Pr = 0.71$, $rr = 2.6$ and Rayleigh numbers of $10^2, 10^3, 10^4, 5\times10^4$. The results of RBF-DQ method [15] as well as the results of Shu [19] are also included in the table for comparison.

DOI: 10.1002/fld

Table III. Comparison of average equivalent heat conductivity.

Methods	$Ra = 10^2$		$Ra = 10^3$		$Ra = 10^4$		$Ra = 5 \times 10^4$	
	$\overline{k}_{\text{eqi}}$	$\overline{k}_{\text{eqo}}$	$\overline{k}_{\text{eqi}}$	$\overline{k}_{\text{eqo}}$	$\overline{k}_{\text{eqi}}$	$\overline{k}_{\text{eqo}}$	$\overline{k}_{\text{eqi}}$	$\overline{k}_{\text{eqo}}$
iRBF-DQ with polynomial	1.001	1.001	1.082	1.082	1.977	1.977	2.957	2.957
iRBF-DQ without polynomial	1.001	1.001	1.082	1.082	1.977	1.977	2.956	2.956
RBF-DQ [15]	1.001	1.001	1.082	1.082	1.976	1.976	2.953	2.954
Reference [19]	1.001	1.001	1.082	1.082	1.979	1.979	2.958	2.958

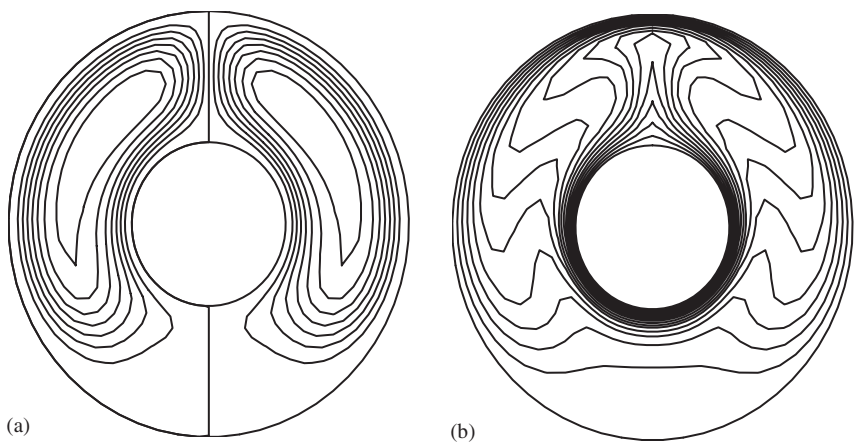

Figure 3. Streamlines and isotherms for natural convection in a concentric annulus obtained by iRBF-DQ without polynomial ($Ra = 5 \times 10^4$, $Pr = 0.71$, $rr = 2.6$): (a) streamlines; and (b) isotherms.

The results of Shu [19] were obtained by using the polynomial based differential quadrature (PDQ) and FDQ methods [2]. They are from the grid-independent study, and can be considered as the benchmark solution. The mesh size used in the present computation is 61×17 (61 in the θ direction and 17 in the r direction). Note that the number of mesh points used in the θ direction is much larger than that used in the r direction. This is because the RBF-DQ is a global method and it can obtain very accurate results by using a considerably small number of mesh points. In contrast, the second-order difference scheme used in the θ direction is a low-order method. To achieve the same order of accuracy as the RBF-DQ method, a much larger number of grid points is needed in the θ direction. It can be obviously observed from Table III that the present results of both iRBF-DQ versions agree very well with the benchmark solution of Shu [19]. The present results are more accurate than those of the original RBF-DQ method [15]. The computed average equivalent conductivities for the inner and outer cylinders are the same. This confirms the theoretical analysis. Since there is no energy loss in the whole system, the theoretical average equivalent conductivities for the inner and outer cylinders should be the same. The flow patterns obtained by both iRBF-DQ versions are the same. Figure 3 shows the streamlines and the isotherms of $Ra = 5 \times 10^4$ obtained by iRBF-DQ without polynomial term.

DOI: 10.1002/fld

It is also found in this paper that the present iRBF-DQ method requires the same level of computational efforts (such as memory requirement and CPU time) as the RBF-DQ method does if the same mesh size is used. It has been proven in our previous paper [20] that the RBF-DQ method, as a kind of global method, has much better computational efficiency as compared with PDQ method [2]. Therefore, we conclude that the present iRBF-DQ method is better than the polynomial based global methods such as PDQ in terms of efficiency. This is because it overcomes the 'oscillation' problems which are inherent in high-order polynomial approximation schemes.

5. CONCLUSIONS

The iRBF-DQ method, which uses iRBFs to construct an N-dimensional linear vector space V_N to obtain the weighting coefficients of derivative approximation, is presented in this paper. The concept of the iRBF is from the work of Nam and Tranh [16, 17]. The major advantage of the iRBF-DQ method is its easy implementation, and flexibility of choosing different iRBFs to construct different iRBF-DQ versions. From solution of one-dimensional Burger's equation and simulation of natural convection in a concentric annulus by solving Navier–Stokes equations, it was found that the numerical results obtained by the iRBF-DQ method agree very well with exact value or available data in the literature. The numerical examples showed that the present method is more accurate than the original RBF-DQ method, while its computational efficiency is kept the same level. It can be recommended that the iRBF-DQ is a good candidate for an accurate approximation of a function and its derivatives, as well as for solving PDE, such as Navier–Stokes equations.

NOMENCLATURE

a_{ij}, b_{ij}	iRBF-DQ weighting coefficients of the first and second-order derivatives
c	shape parameter in RBF approximation
$c0$	dimensionless shape parameter
C_1, C_2	constants of integration
C_p	specific heat at constant pressure
g	gravitational acceleration
$H, \overline{H}$	integrated radial basis functions
k	thermal conductivity
$\overline{k}_{\mathrm{eqi}}, \overline{k}_{\mathrm{eqo}}$	average equivalent heat conductivity for inner cylinder and outer cylinder
Pr	Prandtl number, $Pr = \mu C_p / k$
rr	radius ratio, $rr = L/2R$
R	radius of inner cylinder
Ra	Rayleigh number, $Ra = C_p \rho_0 g \beta L^3 (T_\mathrm{i} - T_\mathrm{o})/kv$
T	temperature
$T_\mathrm{i}, T_\mathrm{o}$	dimensionless temperature on the inner and outer cylinders
u	component of velocity in the r direction
v	component of velocity in the θ direction
w_{ij}	weighting coefficients in DQ method
β	thermal expansion coefficient
φ	radial basis function, $\varphi(r, r_i) = \sqrt{(r - r_i)^2 + c^2}$ in MQ-RBF

DOI: 10.1002/fld

λ	coefficient for φ
μ	viscosity
μ_k	coefficient for φ in iRBF-DQ
v	kinematical viscosity
ρ_0	reference density
ω	vorticity
ψ	stream function

Acronyms

DQ	differential quadrature method
FD	finite difference
FE	finite element
FV	finite volume
iRBF-DQ	integrated radial basis functions-based differential quadrature
IRBFN	integrated radial basis function networks
MQ-RBF	multiquadric radial basis function
PDE	partial differential equation
PDQ	polynomial-based differential quadrature
RBF	radial basis function
RBF-DQ	DQ method with RBF as the base function

REFERENCES

1. Bellman R, Kashef BG, Casti J. Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations. *Journal of Computational Physics* 1972; **10**:40–52.
2. Shu C. *Differential Quadrature and its Application in Engineering*. Springer: London, 2000.
3. Franke R. Scattered data interpolation: tests of some methods. *Mathematics of Computation* 1982; **38**:181–200.
4. Kansa ET. Multiquadrics—A scattered data approximation scheme with applications to computational fluid dynamics: surface approximations and partial derivative estimates. *Computers and Mathematics with Applications* 1990; **19**(6–8):127–145.
5. Kansa ET. Multiquadrics—A scattered data approximation scheme with applications to computational fluid dynamics: solutions to parabolic, hyperbolic, and elliptic partial differential equations. *Computers and Mathematics with Applications* 1990; **19**(6–9):147–161.
6. Fasshauer GE. Solving partial differential equations by collocation with radial basis functions. In *Surface Fitting and Multiresolution Methods*, Mehaute AL, Rabut C, Schumaker LL (eds), 1997; 131–138.
7. Driscoll TA, Forberg B. Interpolation in the limit of increasingly flat radial basis functions. *Computers and Mathematics with Applications* 2002; **43**:413–422.
8. Zhang X, Song KZ, Lu MW, Liu X. Meshless methods based on collocation with radial basis functions. *Computational Mechanics* 2000; **26**(4):333–343.
9. Hon YC, Schaback R. On unsymmetric collocation by radial basis functions. *Applied Mathematics and Computation* 2001; **119**:177–186.
10. Wong SM, Hon YC, Li TS. A meshless multi-layer model for a coastal system by radial basis functions. *Computers and Mathematics with Applications* 2002; **43**(3–4):585–605.
11. Chen W, Tanaka M. A meshless, integration-free, and boundary-only RBF technique. *Computers and Mathematics with Applications* 2002; **43**:379–391.
12. Chen CS, Brebbia CA, Power H. Dual reciprocity method using for Helmholtz-type operators. *Boundary Elements* 1998; **20**:495–504.
13. Shu C, Ding H, Yeo KS. Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 2003; **192**:941–954.

DOI: 10.1002/fld

14. Shu C, Ding H, Chen HQ, Wang TG. An upwind local RBF-DQ method for simulation of inviscid compressible flows. *Computer Methods in Applied Mechanics and Engineering* 2005; **194**:2001–2017.
15. Wu YL, Shu C. Development of RBF-DQ method for derivative approximation and its application to simulate natural convection in concentric annuli. *Computational Mechanics* 2002; **29**:477–485.
16. Nam MD, Tranh TC. Numerical solution of differential equations using multiquadric radial basis function networks. *Neural Networks* 2001; **14**(2):185–199.
17. Nam MD, Tranh TC. Approximation of function and its derivatives using radial basis function networks. *Applied Mathematical Modelling* 2003; **27**:197–220.
18. Shu C, Richards BE. Application of generalised differential quadrature to solve two-dimension incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 1992; **15**:791–798.
19. Shu C. Application of differential quadrature method to simulate natural convection in a concentric annulus. *International Journal for Numerical Methods in Fluids* 1999; **30**:977–993.
20. Wu YL, Shu C, Chen HQ, Zhao N. Radial basis function-enhanced domain-free discretization method and its applications. *Numerical Heat Transfer* 2004; **46**(Part B):269–282.

DOI: 10.1002/fld