EVALUATION OF MULTI-ORDER DERIVATIVES BY LOCAL RADIAL BASIS FUNCTION DIFFERENTIAL QUADRATURE METHOD

L. H. Shen K. H. Tseng D. L. Young *

Department of Civil Engineering
National Taiwan University
Taipei, Taiwan 10617, R.O.C.

ABSTRACT

It is difficult to obtain the derivative values from most mesh dependent numerical procedures in general. This study proposes an efficient computational tool to accurately evaluate the multi-order derivatives by the radial basis functions and local differential quadrature (LRBF-DQ) algorithm. Most of the traditional derivative calculations can be only adopted to evaluate the differential values with the regular meshes. Moreover, the traditional numerical schemes are very restricted by the order of the shape functions. The present technique can be applied to both the structured and unstructured meshes as well as meshless numerical algorithms – such as RBFs and LDQ method. In addition, the proposed model can be also used to estimate multi-order or mixed partial derivative values because its test function using RBFs is a multi-order differentiable function. All of the evaluation of derivative results will be compared with the exact solutions and other numerical techniques. Consequently, this study provides an effective algorithm of post process to accurately calculate the multi-order derivative values for any numerical schemes.

Keywords: Multi-order derivatives, Radial basis functions, Local differential quadrature.

1. INTRODUCTION

Highly accurate physical phenomena can be successfully solved by the efficient numerical methods such as finite element method (FEM), but the derivatives are still subject to its shape functions. Taking the shape function in linear element for example; the first order differential values can be evaluated as the constant coefficients from its shape function. However the multi-order derivatives are difficult to obtain directly from the linear element. The results of the first order derivatives are in general not accurate enough. Nevertheless, the most famous and convenient derivative approximation was based on the Taylor's series expansion, for example central difference scheme (Spalding, [1]). Furthermore, when the extended application of the finite difference method (FDM) to solve high-order derivatives, it will easily be trapped by either the ill-conditioned [2] calculation or poor accuracy [3]. Other researches were introduced to estimate derivatives by polynomial interpolation. These interpolation formulations include polynomials (Slazer, [4]), Hermite, and Lagrange polynomials (Ciarlet and Raviat, [5]; Gasca and Maeztu, [6]). Furthermore, Nikiforov and Sulov, [7] collected and arranged several classical orthogonal polynomials of a discrete variable, which are the difference analogues of the Jacobi, Laguerre, and Hermite polynomials on uniform or non-uniform grids. The derivatives can be numerically calculated by the complex-step derivative formula [8] as well. Lyness and Moler in [9] used complex variables for numerical differentiation. A method of complex Laplace transform based on numerical inversion was developed by Abate and Dubner [10].

Unfortunately most of the above methods have to be adopted with the structured grids, since it is not a common choice in numerical simulation because of mainly irregular domains. Therefore a mesh free post process to cover structured, unstructured, and hybrid meshes for irregular domains was proposed to estimate the multi-order partial derivative values from scattered data. Nayroles *et al*. [11] proposed a diffuse elements method to estimate the derivatives from the physical characteristics, which have been solved by the finite element method. Wu and Shu [12] proposed a new development for derivative approximation by radial basis function based differential quadrature (RBF-DQ) method; this scheme gave the possibility to obtain derivative from scattered points. However, there is one uncertain shape parameter needed to choose from their method. Therefore, Tseng *et al*. [13] introduced a residual analysis into the LRBF-DQ method (Shu, [14]) to decide the optimal shape parameter for derivative approximations at each computing node. For Tseng's method, they took the governing equation into consideration as the clue to decide the uncertain shape pa-

* Corresponding author (dlyoung@ntu.edu.tw)

DOI : 10.1017/jmech.2012.121
Copyright © 2013 The Society of Theoretical and Applied Mechanics, R.O.C.
https://doi.org/10.1017/jmech.2012.121 Published online by Cambridge University Press

rameter. However, it still had one drawback for this method. Their method chose one optimal shape parameter for all derivatives, but present scheme can separately choose the optimal shape parameter for each derivatives. Hence, we will give more detailed discussions in the followings.

2. NUMERICAL SCHEME

The conventional LRBF-DQ methods approximate the derivatives with weighted sum of functional values. Therefore, the quality of weighting coefficients obtained from LRBF-DQ methods is influenced by two factors: The assignment of shape parameter and the selection of supporting knots. This investigation will firstly discuss about the supporting knots selection and also introduce a technique to decide the optimal shape parameter for each derivative.

2.1 Local Differential Quadrature Method

Shu *et al*. in [14] firstly introduced the LRBF-DQ method to solve nature convection, lid-driven cavity flow. It is the numerical method to approximate derivative term in the local scheme, as shown in Fig. 1. The formulation of derivative approximation by LDQ method is as follows:

$$f_{\mathbf{x}}^{(m)}(\mathbf{x})\Big|_i = \sum_{j=1}^{N_L} w_{ij}^{(m)} \cdot f(\mathbf{x}_j), \quad \text{for} \quad i = 1, 2, \cdots, N \tag{1}$$

where N is the number of global computational nodes and N_L indicates the local supporting knots. The subscript i denotes the global node index (the interesting point) and j indicates the local index at each supporting knot. The w_{ij} are the weighting coefficients which can be estimated by the concept of DQ method.

Shu introduced the test function by choosing the RBFs, so the functional values, $f(x)$ are substituted by a test function, $\phi(r_j)$ to Eq. (1). r_j denotes as $\| x–x_j \|$, which is the Euclidean norm in the Cartesian coordinate system. $\phi(r_j)$ can be any radial basis functions. Therefore, there are N_L equations and N_L weighting coefficients w_{ij}, and weighting coefficients can be obtained from these N_L algebraic equations.

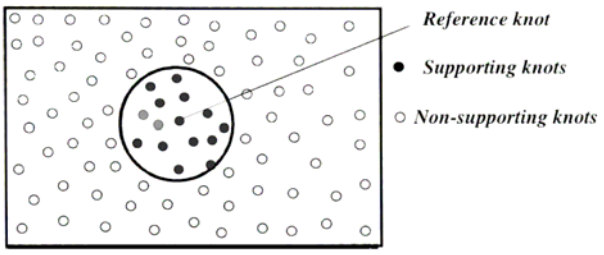


Fig. 1 The conception of grid knots in a LRBF-DQ method [14]

$$\begin{bmatrix} \phi_1(r_1) & \phi_1(r_2) & \cdots & \phi_1(r_N) \\ \phi_2(r_1) & \phi_2(r_2) & & \phi_2(r_N) \\ & & \ddots & \\ \phi_N(r_1) & \phi_N(r_2) & & \phi_N(r_N) \end{bmatrix} \begin{Bmatrix} w_{i1}^{(m)} \\ w_{i2}^{(m)} \\ \vdots \\ w_{iN}^{(m)} \end{Bmatrix} = \begin{Bmatrix} \left.\frac{\partial^m \phi(r_1)}{\partial x^m}\right|_i \\ \left.\frac{\partial^m \phi(r_2)}{\partial x^m}\right|_i \\ \vdots \\ \left.\frac{\partial^m \phi(r_N)}{\partial x^m}\right|_i \end{Bmatrix} \tag{2}$$

where $i = 1, 2, \ldots, N$, and m is the order of derivative in the x direction.

The above Eq. (2) can be evaluated in the same processes for $i = 1, 2, \ldots, N$. Consequently, the weighting coefficients at node i can be solved as follows:

$$\begin{Bmatrix} w_{i1}^{(m)} \\ w_{i2}^{(m)} \\ \vdots \\ w_{iN}^{(m)} \end{Bmatrix} = \begin{bmatrix} \phi_1(r_1) & \phi_1(r_2) & \cdots & \phi_1(r_N) \\ \phi_2(r_1) & \phi_2(r_2) & & \phi_2(r_N) \\ & & \ddots & \\ \phi_N(r_1) & \phi_N(r_2) & & \phi_N(r_N) \end{bmatrix}^{-1} \begin{Bmatrix} \left.\frac{\partial^m \phi(r_1)}{\partial x^m}\right|_i \\ \left.\frac{\partial^m \phi(r_2)}{\partial x^m}\right|_i \\ \vdots \\ \left.\frac{\partial^m \phi(r_N)}{\partial x^m}\right|_i \end{Bmatrix} \tag{3}$$

where $i = 1, 2, \ldots, N$

2.2 Derivative Calculation

We provide a numerical technique to calculate the differential values from the known data. As discussed in the section 2.1, the derivative can be derived from the Eq. (1). In addition, we choose IMQ as the test function to obtain weighting coefficients from the Eq. (3). Its formulation is as follows:

$$\phi(r_j) = (r_j^2 + c^2)^{-1/2} \tag{4}$$

where r_j denotes as $\| x–x_j \|$, which is the Euclidean norm in the Cartesian coordinate system.

Take evaluating u_x in two-dimensional domain for example, the differential test function, $\frac{\partial^m \phi(r_j)}{\partial x^m}$ in Eq. (2) can be replaced by the following formulation,

$$\frac{\partial \phi(r_j)}{\partial x} = \frac{-(x - x_j)}{\left[(x - x_j)^2 + (y - y_j)^2 + c^2\right]^{3/2}} \tag{5}$$

By substituting Eqs. (4) and (5) into Eq. (2), it is obvious to obtain a linear system with N_L equations. Hence, we have to solve the N_L linear algebraic equations as follows:

https://doi.org/10.1017/jmech.2012.121 Published online by Cambridge University Press

$$\begin{bmatrix} \Phi_{11} & \cdots & \Phi_{1N_L} \\ \vdots & \ddots & \vdots \\ \Phi_{N_L1} & \cdots & \Phi_{N_LN_L} \end{bmatrix} \begin{Bmatrix} w_{i1}^{(1)} \\ \vdots \\ w_{iN_L}^{(1)} \end{Bmatrix} = \begin{Bmatrix} \mathrm{D}_1 \\ \vdots \\ \mathrm{D}_{N_L} \end{Bmatrix} \tag{6}$$

where $\Phi_{kj} = \left[(x_k - x_j)^2 + (y_k - y_j)^2 + c^2 \right]^{-1/2}$, $\mathrm{D}_j = \left. \dfrac{\partial \phi(r_j)}{\partial x} \right|_i$ and $w_{ij}^{(1)} = \left[w_{i1}^{(1)}, \cdots, w_{iN_L}^{(1)} \right]^T$ are weighting coefficients for node i and $i \in (1, N)$; $j, k \in (1, N_L)$.

After estimating the weighting coefficients, u_x can be determined directly from the Eq. (1). Based on this concept, we can obtain the weighting coefficients of different derivative values just by replacing the formulation of D_j. The common derivative operators for 2D and 3D formulation are listed in Table 1.

2.3 Supporting Knots Selection

Present technique operates with the weighted sum of the known physical meanings, which have been solved by the numerical schemes. We will choose the supporting knots from the nodes in these numerical schemes. However, the way to select supporting knots will highly influence the quality of the estimation of derivatives. Hence, we will introduce the way to choose supporting knots from structured and unstructured mesh in this section. No matter considering supporting knots in the structured or unstructured mesh by LRBF-DQ method, we choose the supporting knots in each direction uniformly to stabilize the results of derivative approximation. Consequently, we will also pick up the supporting knots under this guidance. It is easier to figure out these supporting knots in the structured mesh because we have known its collocation method. There are different ways to choose the supporting knots in the 2D and 3D structured meshes. The supporting knots for 2D and 3D structured meshes will be considered under the rule in Figs. 2 and 3 respectively for inside point, boundary point, and corner point. As can be seen from Figs. 2 and 3, node #1 denotes as the interesting node (reference knot) and nodes #2 to #27 indicate the sequence we chose from the knots (supporting knots).

Comparing with structured mesh, it is difficult to pick up the supporting knots in each direction uniformly from the unstructured mesh. The only information we have is the relationship between the node and the element. Hence, the supporting knots for 2D and 3D unstructured meshes will be considered by the relationship between node and element as shown in Figs. 4 and 5 respectively for inside point and boundary point. Taking Fig. 4 for example, we can choose supporting knots from the relationship between node and element. Firstly, we find out the elements (element #1 to #5) relative to the reference knot (node #1). In addition, we can record the index of relative elements. For example, element #1 with node #1, #2, #3, and element #2 with node #1, #3, #4, and so on. Thirdly, filtering the repetitive index from these elements we define the residual nodes by the 1st layer supporting knots in this article. Furthermore, the 2nd layer supporting knots are similarly found from step 1 to 3 on the basis of 1st layer's nodes. The supporting knots will be not only picked up uniformly but also chosen under the same concept to any nodes in unstructured and hybrid meshes. Actually, the 1st layer supporting knots can automatically be found from the stiffness matrix during the numerical simulation of FEM. It is completely the same concept to extend to 3D unstructured or hybrid mesh.

Table 1 Differential IMQ formulation operators

$\mathbf{D}_j$	2D formulation	3D formulation
$\dfrac{\partial \phi(r_j)}{\partial x}$	$\dfrac{-(x-x_j)}{(r_{1j}^2 + c^2)^{3/2}}$	$\dfrac{-(x-x_j)}{(r_{2j}^2 + c^2)^{3/2}}$
$\dfrac{\partial \phi(r_j)}{\partial y}$	$\dfrac{-(y-y_j)}{(r_{1j}^2 + c^2)^{3/2}}$	$\dfrac{-(y-y_j)}{(r_{2j}^2 + c^2)^{3/2}}$
$\dfrac{\partial \phi(r_j)}{\partial z}$	None	$\dfrac{-(z-z_j)}{(r_{2j}^2 + c^2)^{3/2}}$
$\dfrac{\partial^2 \phi(r_j)}{\partial x^2}$	$\dfrac{-1}{(r_{1j}^2 + c^2)^{3/2}} + \dfrac{3(x-x_j)^2}{(r_{1j}^2 + c^2)^{5/2}}$	$\dfrac{-1}{(r_{2j}^2 + c^2)^{3/2}} + \dfrac{3(x-x_j)^2}{(r_{2j}^2 + c^2)^{5/2}}$
$\dfrac{\partial^2 \phi(r_j)}{\partial y^2}$	$\dfrac{-1}{(r_{1j}^2 + c^2)^{3/2}} + \dfrac{3(y-y_j)^2}{(r_{1j}^2 + c^2)^{5/2}}$	$\dfrac{-1}{(r_{2j}^2 + c^2)^{3/2}} + \dfrac{3(y-y_j)^2}{(r_{2j}^2 + c^2)^{5/2}}$
$\dfrac{\partial^2 \phi(r_j)}{\partial z^2}$	None	$\dfrac{-1}{(r_{2j}^2 + c^2)^{3/2}} + \dfrac{3(z-z_j)^2}{(r_{2j}^2 + c^2)^{5/2}}$
$\dfrac{\partial^2 \phi(r_j)}{\partial x \partial y}$	$\dfrac{-3(x-x_j)(y-y_j)}{(r_{1j}^2 + c^2)^{5/2}}$	$\dfrac{-3(x-x_j)(y-y_j)}{(r_{2j}^2 + c^2)^{5/2}}$
$\dfrac{\partial^2 \phi(r_j)}{\partial x \partial z}$	None	$\dfrac{-3(x-x_j)(z-z_j)}{(r_{2j}^2 + c^2)^{5/2}}$
$\dfrac{\partial^2 \phi(r_j)}{\partial y \partial z}$	None	$\dfrac{-3(y-y_j)(z-z_j)}{(r_{2j}^2 + c^2)^{5/2}}$
Note: $r_{1j}^2 = (x-x_j)^2 + (y-y_j)^2$ and $r_{2j}^2 = (x-x_j)^2 + (y-y_j)^2 + (z-z_j)^2$		

2.4 Shape Parameter

As mentioned in the section 2.2, we can choose the inverse multi-quadrics (IMQ) function as our test function for the LDQ method. This kind of test function

https://doi.org/10.1017/jmech.2012.121 Published online by Cambridge University Press

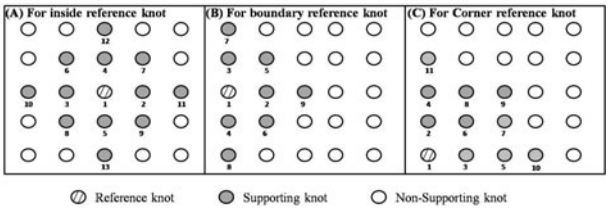


Fig. 2 Diagram of choosing the supporting knots in the 2D structured mesh

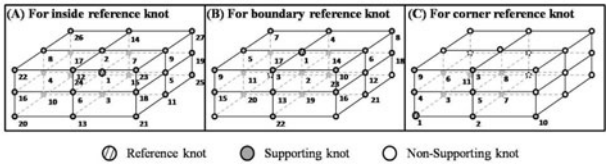


Fig. 3 Diagram of choosing the supporting knots in the 3D structured mesh

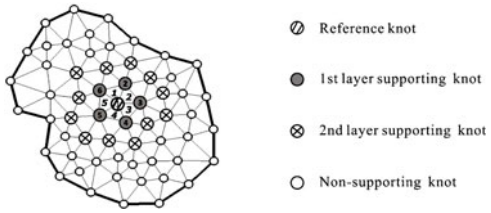


Fig. 4 Diagram of choosing the supporting knots in the 2D unstructured mesh

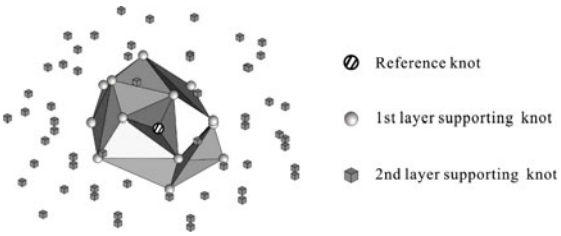


Fig. 5 Diagram of choosing the supporting knots in the 3D unstructured mesh

brings us a great benefit to evaluate multi-order derivatives in any kind of domains, but there is still a free shape parameter, c needed to decide. Moreover, the shape parameter, c has a great influence on the accuracy of derivatives in the process of LRBF-DQ method. Therefore, we will introduce another alternative to decide the optimal shape parameter. There are N_L equations and N_L unknown weighting coefficients in Eq. (6). For each shape parameter, it can be solved using the algorithm system in Eq. (6) to obtain one group of weighting coefficients. The method to obtain the optimal shape parameter is from the characteristic of the sum of weighting coefficients. Taking evaluation of first order derivative with central difference method for example, we can obtain its formulation as follows:

$$\frac{\partial f}{\partial x} \approx \frac{f(x+h)-f(x-h)}{2h} \tag{7}$$

In the beginning of finite difference method, we assume that even if the functional value, $f(x)$ was a constant value, the approximation of the first order derivative should definitely satisfy $\partial f / \partial x = 0$. From Eq. (7), it is obvious that the sum of weighting coefficient must satisfy $\sum w_{ij}^{(1)} = 0$. Hence, we introduce $\sum w_{ij}^{(m)} = 0$ to decide the optimal shape parameter in this study. The residual of the sum of weighting coefficients is denoted by the following formulation.

$$\mathrm{R}(c) = \left| \sum w_{ij}^{(m)} \right| \tag{8}$$

We give a test region of shape parameter to determine each partial differential weighting coefficients from Eq. (6). Moreover, each shape parameter will get one group of weighting coefficients and we will be able to decide the optimal shape parameter while the minimum R(c) occurs.

3. NUMERICAL RESULTS OF DERIVATIVE APPROXIMATIONS

In this section, one simple case will be introduced as an example to discuss the influence of supporting knots and shape parameter while estimating multi-order derivatives by the LRBF-DQ method. The numerical results of derivatives approximation will be compared with the exact solution. Consequently, the definition of errors will be shown at first. Secondly, in order to exclude the influence of shape parameter, we will use the same shape parameter and choose different local supporting knots number to see its influence to derivatives approximation. On the other side, in order to exclude the influence of supporting knots, several constant shape parameters for whole domain will be used to estimate derivatives under the same supporting knots. Furthermore, these results with different shape parameters will be compared with exact solutions and finite element method. Thirdly, we will provide present technique to compare with the optimal results in the above situations. Finally, two testing cases for 2D and 3D domains will be introduced as our verification.

3.1 Error Definition

In this investigation, the maximum error in the whole domain (M.E.) and root mean square error (RMSE) are denoted as the following forms,

$$\mathrm{M.E.} = \mathrm{Max}(|u_{\mathrm{N}i} - u_{\mathrm{A}i}|) \quad i \in (1, N) \tag{9}$$

$$\mathrm{RMSE} = \left(\sum_{i=1}^{\mathrm{N}} |u_{\mathrm{N}i} - u_{\mathrm{A}i}|^2 / N\right)^{1/2} \qquad i \in (1, N) \tag{10}$$

where u_{N} are numerical results, u_{A} are exact results, and N is the number of nodes in the computing domain.

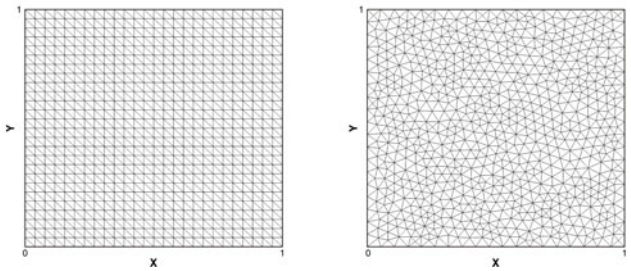
3.2 Testing Case with Square Domain

One differentiable exact solution and its differential values, shown in Table 2, will be considered as example to discuss the influence of choosing the supporting knots. There are two kinds of tested meshes, the structured and unstructured meshes (Fig. 6).

https://doi.org/10.1017/jmech.2012.121 Published online by Cambridge University Press

Table 2 Exact solution for testing case

Form	The formulation of exact solution
u	$e^x \sin x \cos y$
u_x	$e^x \sin x \cos y + e^x \cos x \cos y$
u_y	$-e^x \sin x \sin y$
u_{xx}	$e^x \sin x \cos y + 2e^x \cos x \cos y - e^x \sin x \cos y$
u_{yy}	$-e^x \sin x \cos y$
u_{xy}	$-e^x \sin y (\sin x + \cos x)$



(a) Structured mesh (729 nodes) (b) Unstructured mesh (756 nodes)

Fig. 6 Two kinds of tested meshes

3.2.1 The Influence of Supporting Knots

The supporting knots will be chosen by considering its nearest distance from the reference knot. For example, the number of supporting knots, which denotes as N_L in Figs. 7 and 8 means that they automatically choose N_L nearest knots from that reference knot. We define the above idea by choosing the supporting knots as "automatic selection" in this study. The results of multi-order derivatives of structured and unstructured meshes for different N_L are shown in Figs. 7 and 8, respectively.

From the numerical results in Figs. 7 and 8, we can see that the accuracy of numerical results can be improved by increasing the local supporting knots, especially for the unstructured mesh. Furthermore, the weighting coefficients play an important role to the accuracy of derivative approximation. As shown in Fig. 7, the accuracy of multi-order derivatives are not improved obviously by increasing the number of supporting knots from 9 to 13, because the supporting knots give little contribution from those weighting coefficients in the uncertain direction. For example, the supporting knots in Fig. 9(A) will not actually choose useful supporting knots in the concerned direction from 10 to 13. However, it will be much easier to choose contributive supporting knots for the unstructured mesh, shown in Fig. 9(B). Consequently, while choosing supporting knots with the conception "automatic selection", LRBF-DQ method would get much better derivative approximation under the unstructured mesh than the structured one.

3.2.2 The Influence of Shape Parameter

The multi-order derivatives will be evaluated with different constant shape parameters (c = 0.5 ~ 10) for

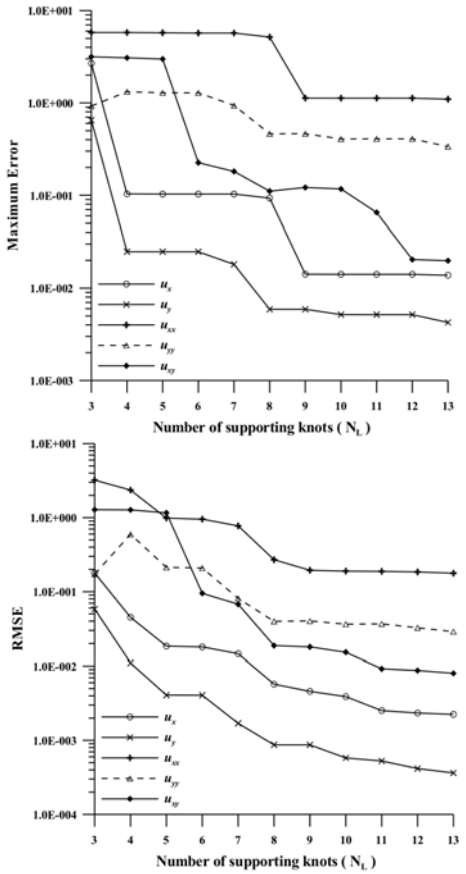


Fig. 7 Relationship between number of supporting knots and errors (Structured mesh)

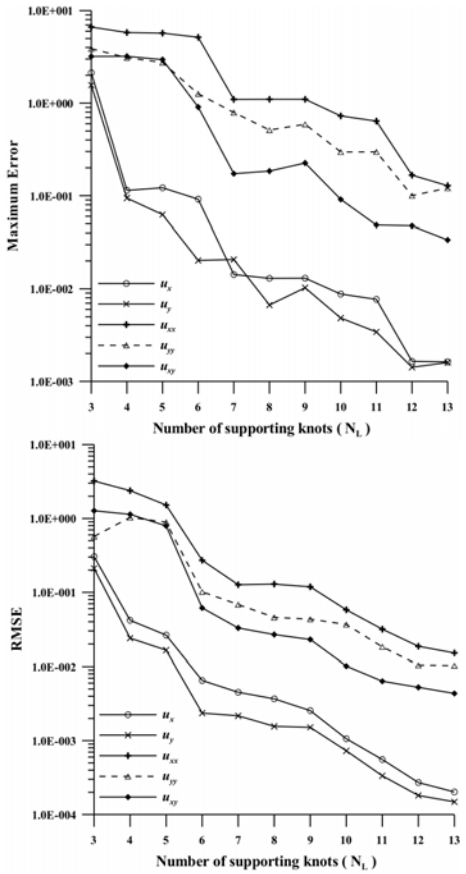


Fig. 8 Relationship between number of supporting knots and errors (Unstructured mesh)

https://doi.org/10.1017/jmech.2012.121 Published online by Cambridge University Press

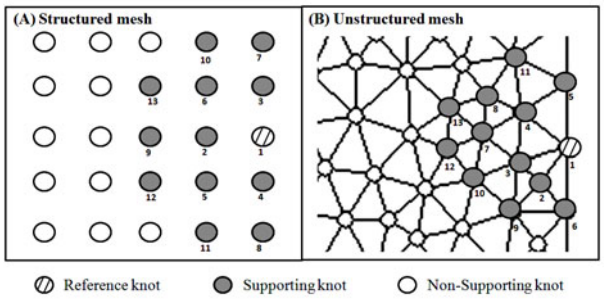


Fig. 9 Diagram for choosing supporting knots by automatic selection

the whole domain points and tested under two different mesh types, as shown in Fig. 2. Moreover, in order to limit the influence of the number of supporting knots, we will choose 13 supporting knots with the concept of "automatic selection". The influence of using various constant shape parameters to obtain multi-order derivatives by LRBF-DQ method for the structured and unstructured meshes is shown in Figs. 10 and 11, respectively.

From the conjuncture revealing in Figs. 10 or 11, we can see that the numerical results by LRBF-DQ method fluctuate with different shape parameters. Hence, we can figure out that the shape parameter plays an important role for evaluating the multi-order derivatives, especially for the unstructured mesh. For example, it can be found in Fig. 10, that the maximum error of U_{xx} in the whole domain can be varied from 0.016 to 16.340 for different shape parameters. Therefore, the methods for deciding optimal shape parameter will be necessary before operating LRBF-DQ method to evaluate the multi-order derivatives.

3.2.3 Comparison of Proposed Method and Conventional LRBF-DQ Method

Present technique for deciding the optimal shape parameter will be used to compare with the best results in the section 3.3. With the same situation in the section 3.3, we will use the supporting knots with the concept of "automatic selection" to estimate multi-order derivatives by present technique. The proposed method is to decide the optimal shape parameter while the minimum residual of the sum of weighting coefficients occurs. The proposed method will be compared with the results of using the constant shape parameter for the whole domain, which are c = 2.0 for structured mesh and c = 4.0 for the unstructured one. The comparisons of derivatives with different methods for the structured and unstructured meshes are shown in Tables 3 and 4, respectively.

Several conclusions can be drawn from Tables 3 and 4. First of all, some results of using constant shape parameter are better than those of present techniques. However, results of using constant shape parameter are chosen on the situation where the derivative is known. While applying LRBF-DQ method to unknown derivative approximation, present two methods give us a benefit to decide the optimal shape parameter automatically with clues rather than just from guess. Secondly, the influence of supporting knots selection can

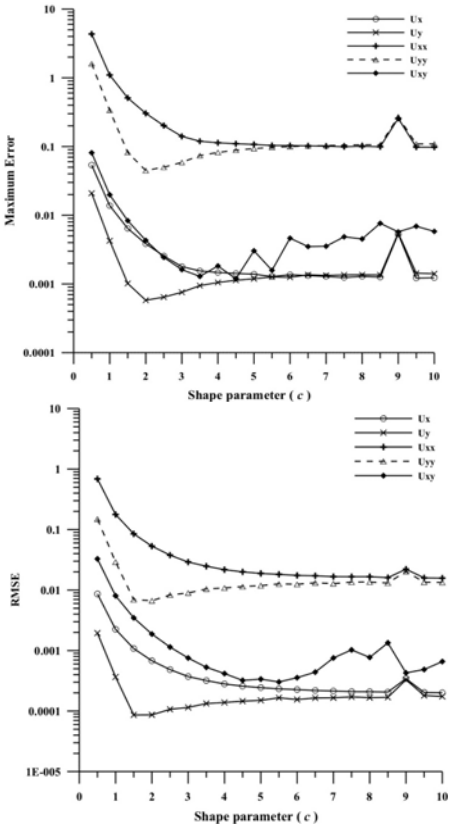


Fig. 10 Relationship between number of shape parameter and errors (Structured mesh)

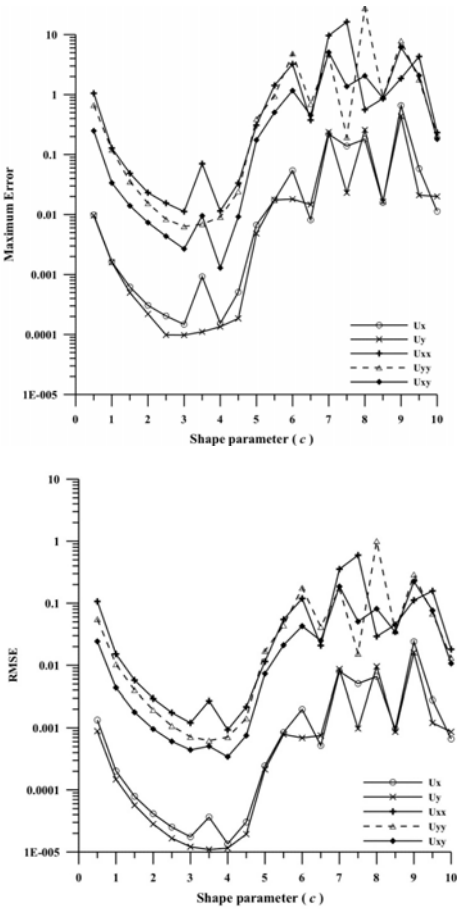


Fig. 11 Relationship between number of shape parameter and errors (Unstructured mesh)

https://doi.org/10.1017/jmech.2012.121 Published online by Cambridge University Press

Table 3 Error analysis of the derivatives with different methods (Structured mesh)

Derivative	Constant $c = 2.0$		Proposed method	
	(MAX-Error)	(RMS-Error)	(MAX-Error)	(RMS-Error)
u_x	3.84×10^{-3}	6.79×10^{-4}	3.39×10^{-3}	6.12×10^{-4}
u_y	5.79×10^{-4}	8.66×10^{-5}	5.54×10^{-4}	8.69×10^{-5}
u_{xx}	0.304	5.32×10^{-2}	0.267	4.79×10^{-2}
u_{yy}	4.47×10^{-2}	6.67×10^{-2}	4.28×10^{-2}	6.72×10^{-3}
u_{xy}	2.44×10^{-3}	1.88×10^{-3}	4.21×10^{-3}	6.77×10^{-4}

Table 4 Error analysis of the derivatives with different methods (Unstructured mesh)

Derivative	Constant $c = 2.0$		Proposed method	
	(MAX-Error)	(RMS-Error)	(MAX-Error)	(RMS-Error)
u_x	1.53×10^{-4}	1.36×10^{-5}	1.06×10^{-4}	1.32×10^{-4}
u_y	1.35×10^{-4}	1.16×10^{-5}	1.05×10^{-4}	1.07×10^{-5}
u_{xx}	1.15×10^{-2}	9.28×10^{-4}	8.15×10^{-3}	8.77×10^{-4}
u_{yy}	9.09×10^{-3}	7.11×10^{-4}	6.41×10^{-3}	6.55×10^{-4}
u_{xy}	1.29×10^{-3}	3.43×10^{-4}	1.35×10^{-3}	3.64×10^{-4}

also be found in Tables 3 and 4. When it comes to the contribution of useful weighting coefficients in the concerned direction, the unstructured mesh will get the better quality of the weighting coefficients than the structured one.

3.2.4 Comparison of Proposed Method and Finite Element Method

Before testing present method with exact solution, we will choose finite element method as an example to compare with two present methods. In general, we operate the FEM scheme with the linear element (Fig. 12) and it approximates the physical detail by the shape function with the following formulation.

$$\hat{u} \approx \sum_{j=1}^{3} \varphi_j u_j \tag{11}$$

From the conception of FEM, it assumes that u_1 to u_3 in Fig. 12 satisfy the function as follows:

$$\begin{aligned} u_1 &= a + bx_1 + cy_1 \\ u_2 &= a + bx_2 + cy_2 \\ u_3 &= a + bx_3 + cy_3 \end{aligned} \tag{12}$$

From Eq. (12), a, b, c can be solved with the following formulation.

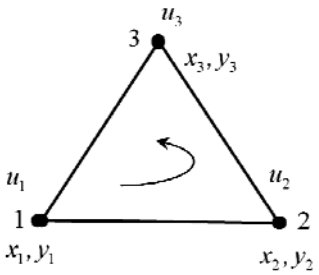


Fig. 12 Diagram of finite element method

$$a = \frac{\begin{vmatrix} u_1 & x_1 & y_1 \\ u_2 & x_2 & y_2 \\ u_3 & x_3 & y_3 \end{vmatrix}}{2\Delta}; \quad b = \frac{\begin{vmatrix} 1 & u_1 & y_1 \\ 1 & u_2 & y_2 \\ 1 & u_3 & y_3 \end{vmatrix}}{2\Delta};$$
$$c = \frac{\begin{vmatrix} 1 & x_1 & u_1 \\ 1 & x_2 & u_2 \\ 1 & x_3 & u_3 \end{vmatrix}}{2\Delta} \text{ and } \Delta = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} \tag{13}$$

where the symbol Δ denotes as the area of the triangle element in Fig. 5.

After substituting Eq. (13) into Eq. (12) and comparing with the Eq. (11), we can obtain the shape function of physical detail as following formulation.

$$\begin{aligned} \varphi_1 &= \left[(x_2 y_3 - x_3 y_2) + (y_2 - y_3)x + (x_3 - x_2)y\right]/2\Delta \\ \varphi_2 &= \left[(x_3 y_1 - x_1 y_3) + (y_3 - y_1)x + (x_1 - x_3)y\right]/2\Delta \\ \varphi_3 &= \left[(x_1 y_2 - x_2 y_1) + (y_1 - y_2)x + (x_2 - x_1)y\right]/2\Delta \end{aligned} \tag{14}$$

Therefore, we can derive the first order of derivative with respect to x or y in the following formulation.

$$\left(\frac{\partial \hat{u}}{\partial x_k}\right)_{ei} \approx \sum_{j=1}^{3} \frac{\partial \varphi_j}{\partial x_k} u_j \qquad k \in (1, 2) \tag{15}$$

However, the derivatives in the above formulation represent the element, *i*, but all we concern are the derivative at the node. Hence, we will approximate the derivative by considering the average of its nearby elements in the following formulation.

$$\left(\frac{\partial \hat{u}}{\partial x_k}\right)_i \approx \left(\sum_{j=1}^{\mathrm{NR}} \frac{\partial \hat{u}}{\partial x_k}\right)_j A_j \Big/ \sum_{j=1}^{\mathrm{NR}} A_j \qquad k \in (1, 2); \quad i \in (1, N) \tag{16}$$

where NR denotes as the number of relative element to node, *i*, and *N* denotes as the total number of global node.

The above concept is how we obtain the first order derivative from the FEM in the following case. Here, we will also use the same situation in the previous subsection as example to compare with proposed method and it will be tested with the unstructured mesh in Fig. 2(B). The comparison of FEM and present method is shown in Table 5.

https://doi.org/10.1017/jmech.2012.121 Published online by Cambridge University Press

Table 5 Error analysis of the derivatives with different methods

Error	Derivative	FEM	Proposed method
M.E.	u_x	7.28×110^{-2}	1.06×110^{-4}
	u_y	7.37×110^{-2}	1.05×110^{-4}
	u_{xx}	None	8.15×110^{-3}
	u_{yy}	None	6.41×110^{-3}
	u_{xy}	None	1.35×110^{-3}
RMSE	u_x	4.36×110^{-4}	1.32×110^{-4}
	u_y	2.97×110^{-4}	1.07×110^{-5}
	u_{xx}	None	8.77×110^{-4}
	u_{yy}	None	6.55×110^{-4}
	u_{xy}	None	3.64×110^{-4}

As shown in Table 5, we can see that present method is shown with the better results than the FEM. Furthermore, the FEM is limited to its shape function with linear element so that the highest order of derivatives can only be evaluated at the first order. However, two present methods are not only better than the FEM but also can generally apply to evaluate multi-order derivatives.

3.3 Testing Cases with Unstructured Mesh

The common numerical scheme is operated under the unstructured or hybrid mesh, so that present techniques will be tested with these two meshes in this section. The testing cases and mesh information in the 2D and 3D domains will be introduced at first. And then the numerical results will be tested with different global nodes to see if it will improve the derivative approximation. All of them will be estimated by present numerical technique in this section.

We design an exact solution in this section. The case study for 2D and 3D domains are shown in Tables 6 and 7, respectively.

There are three mesh types for 2D and 3D cases as example in this section. The mesh information is shown in Figs. 13 and 14, respectively. We choose coarse and dense unstructured and one boundary layer hybrid meshes as 2D example in Fig. 13. Limiting to the capability of generating the hybrid mesh in the 3D domain, we just set up coarse and dense as well as non-uniformly unstructured meshes as 3D example in Fig. 14.

The numerical results of derivative approximation by the proposed method will separately be compared with the exact solution. Furthermore, they will be tested under three mesh types in Fig. 13; the mesh types are coarse and dense unstructured meshes and one boundary layer typed mesh. The boundary layer typed mesh is sometimes used for dealing with problem in the numerical scheme, so we also operate this kind of mesh type by present technique in this section. The comparison of the derivatives approximation by present method under different mesh types are shown in Tables 8 and 9 for 2D and 3D cases, respectively.

Table 6 Exact solution for 2D case

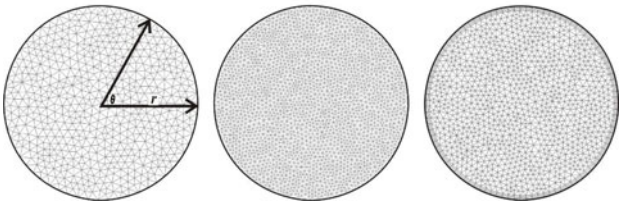
Form	The formulation of exact solution
u	$R^2 - R^8$
u_x	$2x - 8xR^6$
u_y	$2y - 8yR^6$
u_{xx}	$2 - 48x^2R^4 - 8R^6$
u_{yy}	$2 - 48y^2R^4 - 8R^6$
u_{xy}	$-48xyR^2$

Note: $R = \sqrt{x^2 + y^2}$

Table 7 Exact solution for 3D case

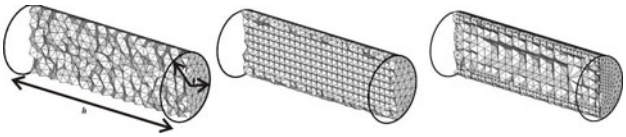
Form	The formulation of exact solution
u	$(2z/3 + 1)(R^2 - R^8)$
u_x	$(2z/3 + 1)(2x - 8xR^6)$
u_y	$(2z/3 + 1)(2y - 8yR^6)$
u_z	$2/3(R^2 - R^8)$
u_{xx}	$(2z/3 + 1.0)(2 - 8R^6 - 48x^2R^4)$
u_{yy}	$(2z/3 + 1.0)(2 - 8R^6 - 48y^2R^4)$
u_{zz}	0

Note: $R = \sqrt{x^2 + y^2}$



(a) Mesh1: 582 nodes (b) Mesh2: 2288 nodes (c) Mesh3: 1451 nodes

Fig. 13 Three kinds of mesh information for 2D testing case ($r = 1$)



(a) Mesh1: 2759 nodes (b) Mesh2: 5051 nodes (c) Mesh3: 8056 nodes

Fig. 14 Three kinds of mesh information for 3D testing case ($r = 0.5$, $h = 3.0$)

3.3.1 2D Case

From Table 8, the proposed method shows that the results can successively be improved by increasing the mesh density. Although the maximum error of u_{xx}, u_{yy}, and u_{xy} terms seem still very large in Table 8, it is because the derivatives on the boundary are with larger magnitudes and hard to obtain. From Figs. 15-1 to -5, it is obvious that the multi-order derivative approximations by present method are improved its accuracy while using more nodes on the boundary (Mesh 2 and Mesh 3.) Furthermore, no matter the coarse or dense meshes are used, the phenomenon of each numerical derivative reflects the same tendency for the exact solutions in Figs. 15-1 to -5.

https://doi.org/10.1017/jmech.2012.121 Published online by Cambridge University Press

Table 8 Error analysis of the derivatives with different meshes for 2D case

Error	Derivative	Mesh 1	Mesh 2	Mesh 3
M.E.	u_x	6.41×10^{-2}	2.73×10^{-3}	1.81×10^{-3}
	u_y	3.25×10^{-2}	6.07×10^{-3}	5.78×10^{-3}
	u_{xx}	2.775	0.210	0.439
	u_{yy}	1.653	0.880	0.438
	u_{xy}	0.822	0.307	0.221
RMSE	u_x	3.39×10^{-3}	1.75×10^{-4}	3.36×10^{-4}
	u_y	2.20×10^{-3}	2.01×10^{-4}	3.90×10^{-4}
	u_{xx}	0.146	1.50×10^{-2}	7.03×10^{-2}
	u_{yy}	0.102	2.32×10^{-2}	7.09×10^{-2}
	u_{xy}	5.82×10^{-2}	1.10×10^{-2}	4.05×10^{-2}

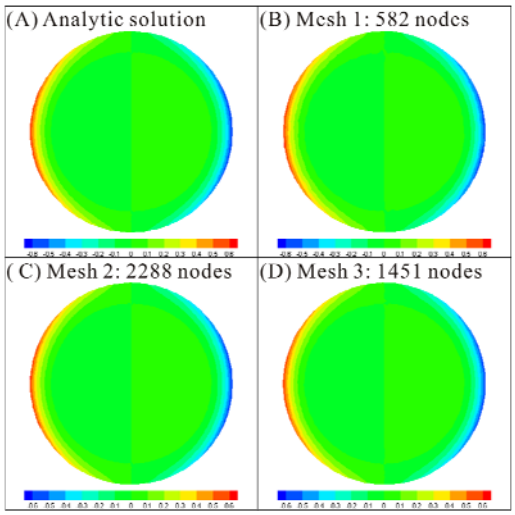


Fig. 15-1 The comparison of u_x term with different mesh types for 2D case

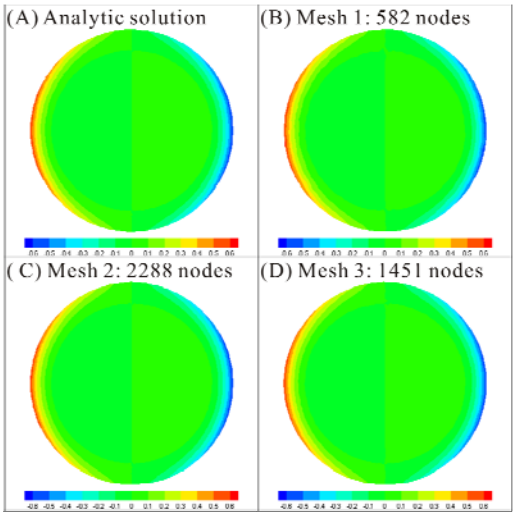


Fig. 15-2 The comparison of u_y term with different mesh types for 2D case

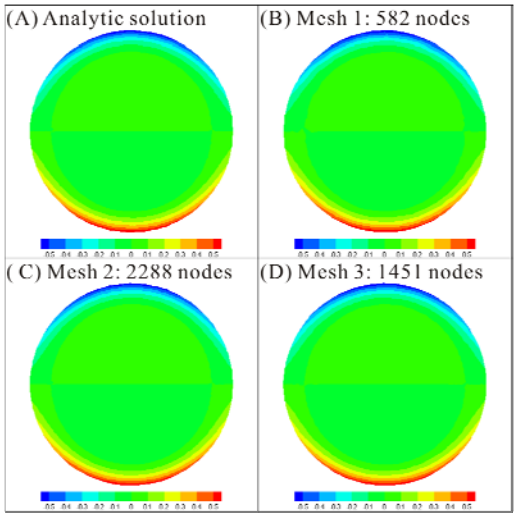


Fig. 15-3 The comparison of u_{xx} term with different mesh types for 2D case

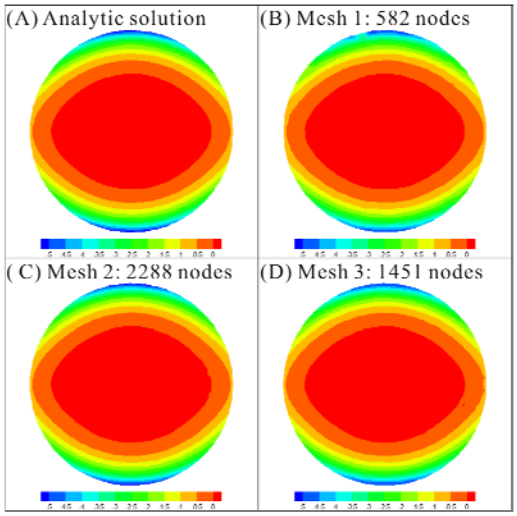


Fig. 15-4 The comparison of u_{yy} term with different mesh types for 2D case

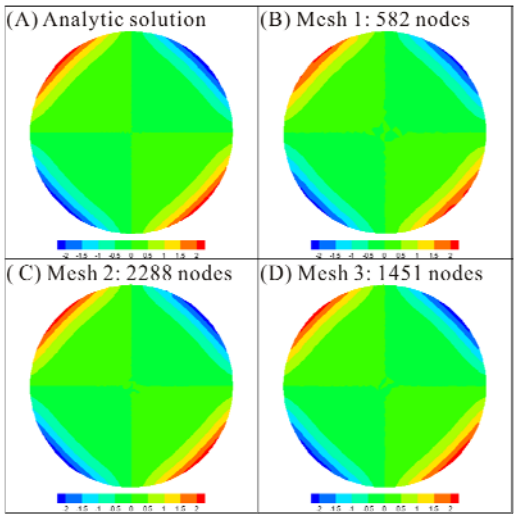


Fig. 15-5 The comparison of u_{xy} term with different mesh types for 2D case

https://doi.org/10.1017/jmech.2012.121 Published online by Cambridge University Press

As shown in Fig. 16, most of the absolute errors for mesh 1 occur near the boundary, it is because the supporting knots in the mesh 1 do not give enough information to approximate the higher order derivatives near the boundary. Hence, one way to improve the accuracy of higher order derivatives near the boundary is by increasing the nodes there. We apply the finer mesh and boundary layer typed mesh in this section and both of them show much better results for estimating the derivatives as shown in Fig. 16.

3.3.2 3D Case

From Table 9, the present method shows that the results can successively be improved by increasing the mesh density. Although the maximum errors of u_{xx}, u_{yy}, and u_{zz} terms seem still large in Table 9, it is because the derivatives on the boundary are with larger resolution and hard to obtain there. From Figs. 17-1 to -6, it is similar to the 2D case that the finer mesh can obtain the better accuracy. Especially by increasing more nodes on the boundary (Mesh 3), we can approximate the derivatives with higher accuracy than those of the coarse one (Mesh 2). Furthermore, no matter the coarse or dense meshes are employed, the phenomenon of each numerical derivative reflects the same tendency for the exact solutions as shown in Figs. 17-1 to -6.

As depicted in Fig. 18, most of the absolute errors for estimating under mesh 1 occur near the boundary, it is because the density of supporting knots does not give enough information to approximate the higher order derivatives there. Furthermore, evaluating derivatives on the boundary will only consider one side of supporting knots so that it will lead to worse results there. Hence, we recommend increasing the nodes or node density near the boundary to improve the accuracy of higher order derivatives, especially for those physical characteristics vary drastically on the boundary.

Table 9 Error analysis of the derivatives with different meshes for 3D case

Error	Derivative	Mesh 1	Mesh 2	Mesh 3
M.E.	T_x	3.60×10^{-2}	1.09×10^{-2}	6.91×10^{-3}
	T_y	2.20×10^{-2}	1.41×10^{-2}	4.79×10^{-3}
	T_z	2.08×10^{-2}	4.37×10^{-3}	3.09×10^{-3}
	T_{xx}	1.123	0.503	0.382
	T_{yy}	0.888	0.655	0.322
	T_{zz}	0.635	0.202	0.148
RMSE	T_{xx}	3.06×10^{-3}	1.05×10^{-3}	3.57×10^{-4}
	T_y	3.07×10^{-3}	1.07×10^{-3}	3.57×10^{-4}
	T_z	8.94×10^{-4}	2.04×10^{-4}	1.04×10^{-4}
	T_{xx}	0.117	5.50×10^{-2}	2.59×10^{-2}
	T_{yy}	0.119	5.53×10^{-2}	2.60×10^{-2}
	T_{zz}	2.98×10^{-2}	8.04×10^{-3}	5.57×10^{-3}

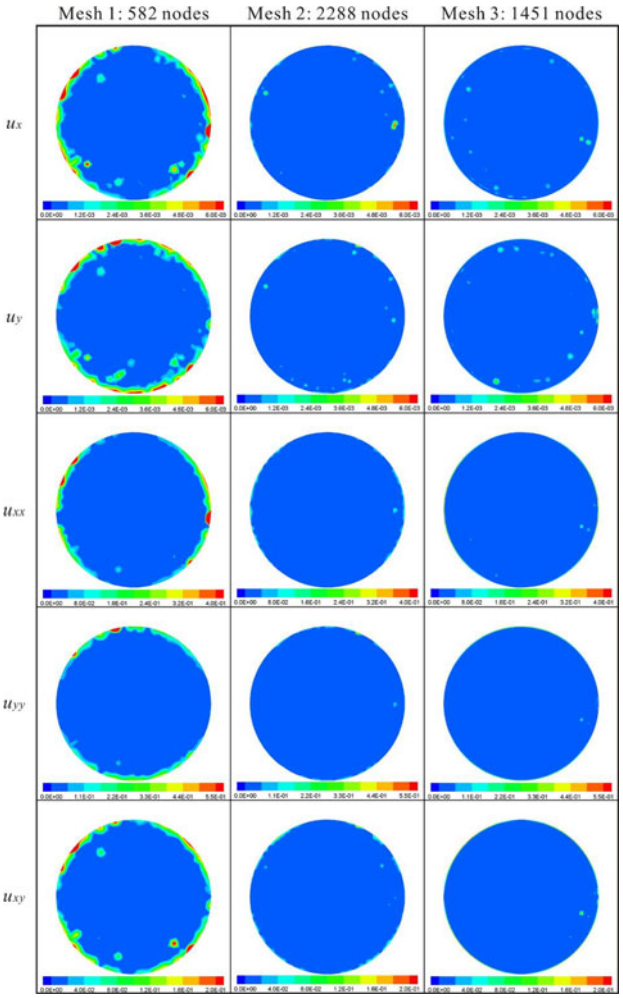


Fig. 16 The error distribution with different mesh types for 2D case

4. CONCLUSION

On the basis of the LDQ method, we have extended this scheme to deal with derivative approximation. From the conception of the DQ method, it is well known that the quality of weighting coefficients highly influences the results of derivative approximation. Furthermore, the quality of weighting coefficients in the DQ method heavily depends on supporting knots selection and the test functions. In this study, we improve the techniques to obtain the better quality of weighting coefficients. On one hand, we can select several mesh types in general numerical scheme to discuss the better way to pick up supporting knots. On the other hand, we can also propose a method to decide the optimal shape parameter by considering the sum of weighting coefficients. We perform some studies to choose the supporting knots in each direction uniformly. Furthermore, we introduced a method to decide the optimal shape parameter. After taking supporting knots selection and the optimal shape parameter into consideration, the quality of weighting coefficients of the LRBF-DQ method are much improved with more stable and accurate results. Based on the better weighting coefficients, we have obtained excellent results of derivative approximations by present method.

https://doi.org/10.1017/jmech.2012.121 Published online by Cambridge University Press

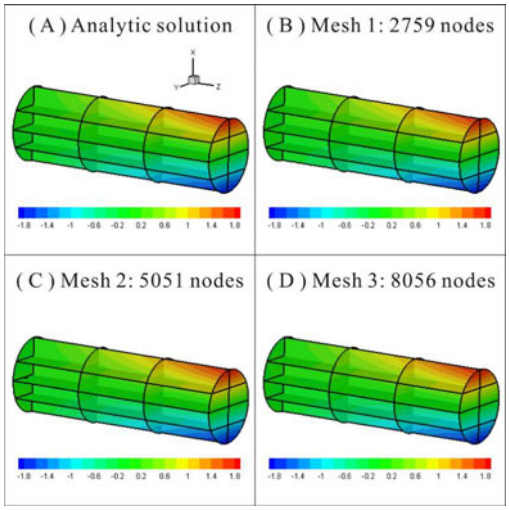


Fig. 17-1 The comparison of u_x term with different mesh types for 3D case

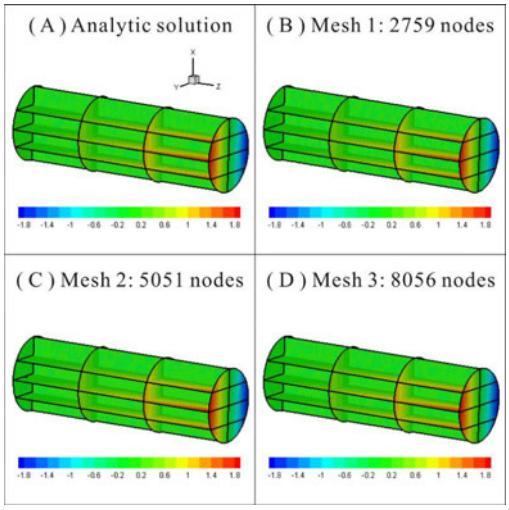


Fig. 17-2 The comparison of u_y term with different mesh types for 3D case

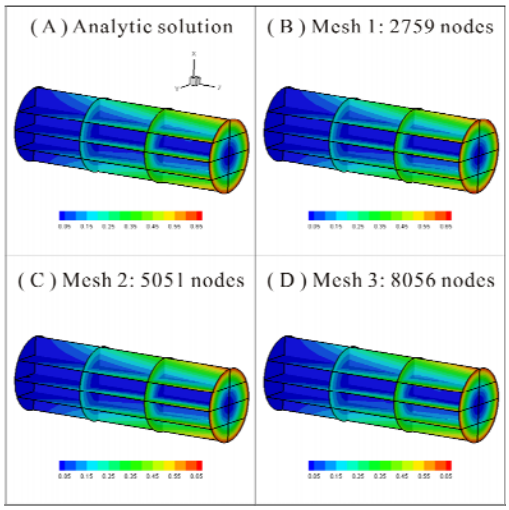


Fig. 17-3 The comparison of u_z term with different mesh types for 3D case

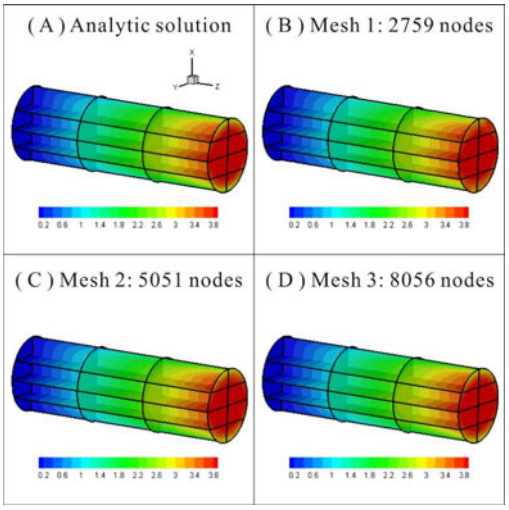


Fig. 17-4 The comparison of u_{xx} term with different mesh types for 3D case

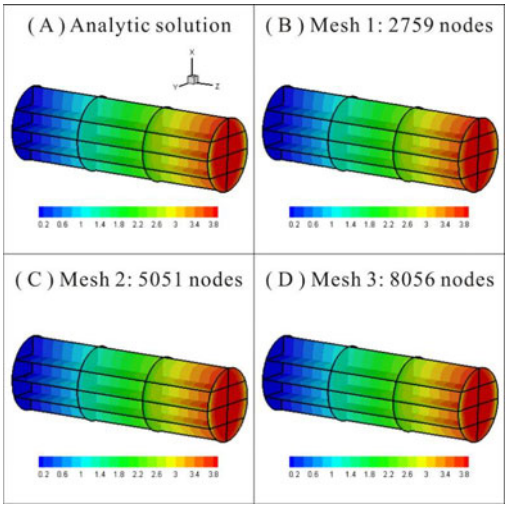


Fig. 17-5 The comparison of u_{yy} term with different mesh types for 3D case

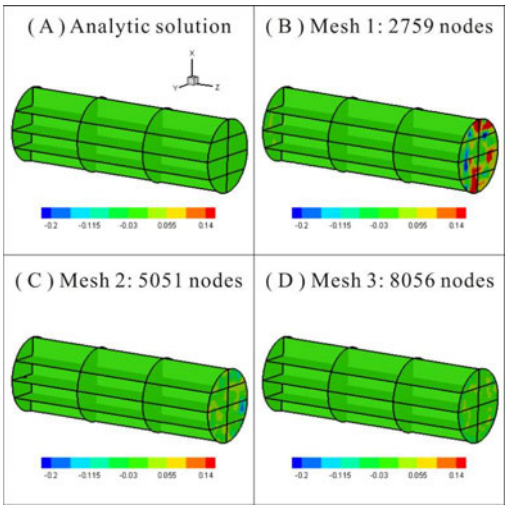


Fig. 17-6 The comparison of u_{zz} term with different mesh types for 3D case

https://doi.org/10.1017/jmech.2012.121 Published online by Cambridge University Press

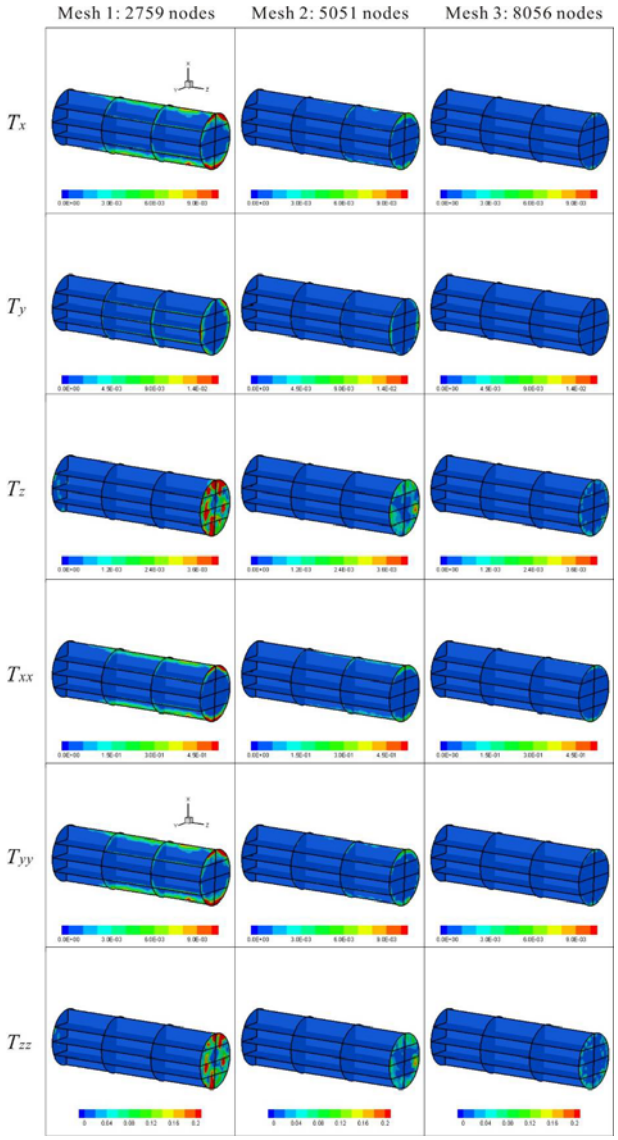


Fig. 18 The error distribution with different mesh types in 3D case

ACKNOWLEDGEMENTS

The writers would like to acknowledge the kind funding support from the National Science Council of Taiwan, through Grants NSC 97-2221-E-002-247-MY3 and NSC 97-2221-E-002-203-MY3.

REFERENCES

1. Spalding, D. B., "A Novel Finite Difference Formulation for Differential Expressions Involving Both First and Second Derivatives," *International Journal for Numerical Methods in Engineering*, **4**, pp. 551–559, (1972).
2. Fornberg, B., "Numerical Differentiation of Analytic Functions," *ACM Transactions on Mathematical Software*, **7**, pp. 512–526 (1981).
3. Squire, W. and Trapp, G., "Using Complex Variables to Estimate Derivatives of Real Functions," SIAM Review (1998).
4. Slazer, H. E., "Divided Differences for Functions of Two Variables for Irregularly Spaced Argument," *Numerische Mathematik*, **6**, pp. 68–77 (1964).
5. Ciarlet, P. G. and Raviat, P. A., "General Lagrange and Hermite Interpolation in Rn with Applications to Finite Element Methods," *Archive for Rational Mechanics and Analysis*, **46**, pp. 177–199 (1972).
6. Gasca, M. and Maeztu, J. I., "On Lagrange and Hermite Interpolation in Rk," *Numerische Mathematik*, **39**, pp. 1–14 (1982).
7. Nikiforov, A. F. and Sulov, S. K., "Classical Orthogonal Polynomials of a Discrete Variable on Nonuniform Lattices," *Letters in Mathematical Physics*, **11**, pp. 27–34 (1985).
8. Martins, J. R. R. A., Sturdza, P. and Alonso, J. J., "The Complex-Step Derivative Approximation," *ACM Transactions on Mathematical Software*, **29**, pp. 245–262 (2003).
9. Lyness, J. N. and Moler, C. B., "Numerical Differentiation of Analytic Functions," *SIAM Journal of Numerical Analysis*, **4**, pp. 202–210 (1967).
10. Abate, J. and Dubner, H., "A New Method for Generating Power Series Expansions of Functions," *SIAM Journal of Numerical Analysis*, **5**, pp. 102–112 (1968).
11. Nayroles, B., Touzot, G. and Villon, P., "Generalizing the Finite Element Method- Diffuse Approximation and Diffuse Elements," *Computational Mechanics*, **10**, pp. 307–318 (1992).
12. Wu, Y. L. and Shu, C., "Development of RBF-DQ Method for Derivative Approximation and its Application to Simulate Natural Convection in Concentric Annuli," *Computational Mechanics*, **29**, pp. 477–485 (2002).
13. Tseng, K. H., Shen, L. H. and Young, D. L., "Evaluating Accurate Differential Derivative by Local Differential Quadrature," *2009 Computational Fluid Dynamics National Conference*, Yilan, Taiwan (2009).
14. Shu, C., H., Yeo, K. S., "Local Radial Basis Differential Quadrature Method and its Application to Solve Two-Dimensional Incompressible Navier-Strokes Equations," *Computer Methods in Applied Mechanics and Engineering*, **192**, pp. 941–954 (2003).

(Manuscript received November 10, 2011, accepted for publication January 10, 2012.)

https://doi.org/10.1017/jmech.2012.121 Published online by Cambridge University Press