


ORIGINAL ARTICLE

Open Access



# New Immersed Boundary Method on the Adaptive Cartesian Grid Applied to the Local Discontinuous Galerkin Method

Xu-Jiu Zhang , Yong-Sheng Zhu\*, Ke Yan and You-Yun Zhang

## Abstract

Currently, many studies on the local discontinuous Galerkin method focus on the Cartesian grid with low computational efficiency and poor adaptability to complex shapes. A new immersed boundary method is presented, and this method employs the adaptive Cartesian grid to improve the adaptability to complex shapes and the immersed boundary to increase computational efficiency. The new immersed boundary method employs different boundary cells (the physical cell and ghost cell) to impose the boundary condition and the reconstruction algorithm of the ghost cell is the key for this method. The classical model elliptic equation is used to test the method. This method is tested and analyzed from the viewpoints of boundary cell type, error distribution and accuracy. The numerical result shows that the presented method has low error and a good rate of the convergence and works well in complex geometries. The method has good prospect for practical application research of the numerical calculation research.

**Keywords:** Immersed boundary method, Adaptive Cartesian grid, Local discontinuous Galerkin method, Reconstruction, Heat transfer equation

## 1 Introduction

The immersed boundary method (IBM) is an effective method for studying complex boundary. Imposing a boundary condition is not straightforward. To solve this problem, different IBMs have been proposed in literature.

Generally, IBMs can be classified into two categories, i.e., the continuous force approach and discrete force approach [1]. The continuous force approach is not suitable for computing the high Reynolds number flows. Therefore, many researchers focus on the discrete force approach. Fadlun et al. [2] implemented the discrete-time forcing approach on a standard marker- and -cell (MAC) staggered grid. Tseng et al. [3] extended the idea of Verzicco et al. [4] and proposed the ghost-cell IBM (GCIBM) for simulating turbulent flows in complex geometries. Mittal et al. [5] used a sharp interface IBM to simulate incompressible viscous flows past three-dimensional immersed bodies. Using the ghost point treatment as a starting point, Gao et al.

[6] improved the method of Tseng et al. [3]. The method effectively eliminates numerical instabilities caused by matrix inversion and flexibly. To improve the accuracy at the boundaries, Shinn et al. [7] implemented the immersed boundary method using the ghost cell approach, whereby the incompressible flows are solved on a staggered grid. To control the spurious force oscillations, Lee et al. [8] proposed a fully-implicit ghost-cell IBM for simulating flows over complex moving bodies on a Cartesian grid. The method is well capable of controlling the generation of spurious force oscillations on the surface of a moving body, thereby producing an accurate and stable solution. To simulate high-Reynolds number compressible viscous flows on adaptive Cartesian grids, Hu et al. [9] present a new ghost-cell turbulent wall boundary condition. In the frame of adaptive Cartesian grids, a cell-centered, second-order accurate finite volume solver has been developed for predicting turbulent flow fields. The robustness and accuracy of the methodology have been validated against well-documented turbulent flow test problems. Now, the IBM method is applied to many fluid dynamic problems such as heat transfer problems [10–12], fluid-solid interaction

\*Correspondence: [yszhu@mail.xjtu.edu.cn](mailto:yszhu@mail.xjtu.edu.cn)  
Key Laboratory of Education Ministry for Modern Design & Rotor-Bearing System, Xi'an Jiaotong University, Xi'an 710049, China

problems [13], complex/moving boundary problems [14], incompressible flows [15], and natural convection problems [16].

The above numerical calculation method adopts the finite difference method, the finite volume method or the finite element method. In this paper, a new IBM to solve the second-order partial equation applied to the local discontinuous Galerkin method (LDG) is presented and we analyzed the causes of error variation for the adaptive Cartesian grid. The LDG method [17–20] means it easy to achieve high accuracy in space and time and provides useful mathematical properties with respect to conservation, stability, and super convergence. In particular, the LDG method can use the mesh with the hanging node [21, 22] for calculation, and it is convenient to apply the method for simulating flows in complex geometries. Therefore, the adaptive Cartesian grid is used for future easier engineering applications [23–25].

This paper organized as follows: In Section 2, we recall the LDG method. In Section 3, the presented IBM including the numerical procedure, definition of ghost flow nodes and the algorithm of reconstruction is described in detail. In Section 4, the results of testing for the accuracy, convergence rate, and effectiveness in complex geometries of the method are presented. Finally, the concluding remarks follows in Section 5.

## 2 Local Discontinuous Galerkin Method

In this study, we tested the method applied to the LDG method on adaptive Cartesian grids for the following classical model elliptic problem:

$$-\Delta u = f, \quad \text{in } \Omega, \quad (1)$$

$$u = g_D, \quad \text{in } \Gamma_D, \quad (2)$$

$$\nabla u \cdot \mathbf{n} = g_N \cdot \mathbf{n}, \quad \text{in } \Gamma_N. \quad (3)$$

where  $\Omega$  is a bounded domain of  $\mathbb{R}^d$ ,  $d = 2, 3$ , and  $\mathbf{n}$  is the outward unit normal to its boundary  $\Gamma = \Gamma_D \cup \Gamma_N$ .

To construct the LDG formulation, we rewrite our elliptic model problem as the following system of first-order equation:

$$\mathbf{q} = \nabla u, \quad \text{in } \Omega, \quad (4)$$

$$-\nabla \cdot \mathbf{q} = f, \quad \text{in } \Omega, \quad (5)$$

$$u = g_D, \quad \text{in } \Gamma_D, \quad (6)$$

$$\mathbf{q} \cdot \mathbf{n} = g_N \cdot \mathbf{n}, \quad \text{in } \Gamma_N. \quad (7)$$

Let us multiply Eqs. (4) and (5) by arbitrary smooth test function  $v$  and  $r$ , respectively, integrate them over an arbitrary element  $\Omega$ , and apply Green's theorem to write

$$\int_{\Omega} \mathbf{q} \cdot \mathbf{r} dx dy = \oint_{\Gamma} u \mathbf{r} \cdot \mathbf{n} ds - \int_{\Omega} u \nabla \cdot \mathbf{r} dx dy, \quad (8)$$

$$\int_{\Omega} \mathbf{q} \cdot \nabla v dx dy = \oint_{\Gamma} v \mathbf{q} \cdot \mathbf{n} ds + \int_{\Omega} f v dx dy. \quad (9)$$

Then we replace the exact solution  $(\mathbf{q}, u)$  by its approximation  $(\mathbf{q}_N, u_N)$  in the element space  $M_N \times V_N$ , where

$$M_N := \left\{ \mathbf{q} \in \left( L^2(\Omega) \right)^d : \mathbf{q}|_K \in S(K)^d, \forall K \in \Gamma \right\}, \quad (10)$$

$$V_N := \left\{ u \in \left( L^2(\Omega) \right)^d : u|_K \in S(K), \forall K \in \Gamma \right\}, \quad (11)$$

$$S(K) := \left\{ \text{polynomials of degree at most } k \text{ in each variables on } K \right\}. \quad (12)$$

The method involves finding  $(\mathbf{q}_N, u_N) \in (M_N \times V_N)$  such that

$$\int_{\Omega} \mathbf{q}_N \cdot \mathbf{r} dx dy = \oint_{\Gamma} \hat{u}_N \mathbf{r} \cdot \mathbf{n}_K ds - \int_{\Omega} u_N \nabla \cdot \mathbf{r} dx dy, \quad (13)$$

$$\int_{\Omega} \mathbf{q}_N \cdot \nabla v dx dy = \oint_{\Gamma} v \hat{\mathbf{q}}_N \cdot \mathbf{n}_K ds + \int_{\Omega} f v dx dy. \quad (14)$$

To complete the definition of the LDG method, we need to introduce some notations. Here,  $\mathbf{n}_K$  denotes the unit outward normal to  $\partial K$ . The functions  $\hat{u}_N$  and  $\hat{\mathbf{q}}_N$  are the numerical fluxes. Let  $K^+$  and  $K^-$  be two adjacent element of  $\Gamma$ ; Let  $x$  be an arbitrary point of the  $(d-1)$  dimension face  $e = K^+ \cap K^-$ , let  $\mathbf{n}^+$  and  $\mathbf{n}^-$  be corresponding outward unit normals at that point. Let  $(\mathbf{q}, u)$  on  $e$  be the interior of  $K^\pm$ . The mean values  $\{\{\cdot\}\}$  and jump  $\llbracket \cdot \rrbracket$  of a function  $u$  and a vector  $\mathbf{q}$  at  $(x, y)$  on an edge  $\Gamma$  are defined as

$$\{\{u\}\} = (u^+ + u^-)/2, \quad \{\{\mathbf{q}\}\} = (\mathbf{q}^+ + \mathbf{q}^-)/2, \quad (15)$$

$$\begin{aligned} \llbracket u \rrbracket &= (u^+ \cdot \mathbf{n}^+ + u^- \cdot \mathbf{n}^-)/2, \quad \llbracket \mathbf{q} \rrbracket \\ &= (\mathbf{q}^+ \cdot \mathbf{n}^+ + \mathbf{q}^- \cdot \mathbf{n}^-)/2. \end{aligned} \quad (16)$$

We are now ready to introduce the expressions that define the numerical fluxes. If  $e$  is inside domain  $\Omega$ , we take

$$\begin{aligned} \hat{\mathbf{q}} &:= \{\{\mathbf{q}\}\} - C_{11} \llbracket u \rrbracket - C_{12} \cdot \llbracket \mathbf{q} \rrbracket, \\ \hat{u} &:= \{\{u\}\} + C_{12} \cdot \llbracket u \rrbracket. \end{aligned} \quad (17)$$

Further, if  $e$  lies on the boundary of  $\Omega$ ,

$$\hat{\mathbf{q}} := \begin{cases} \mathbf{q}^+ - C_{11}(u^+ - g_N)\mathbf{n}, & \text{on } \Gamma_D, \\ \mathbf{g}_N, & \text{on } \Gamma_N, \end{cases} \quad (18)$$

$$\hat{u} := \begin{cases} g_D, & \text{on } \Gamma_D, \\ u^+, & \text{on } \Gamma_N. \end{cases} \quad (19)$$

In the LDG method, stabilization parameter  $C_{11}$  and auxiliary parameter  $C_{12}$  are defined on each face  $e$  as follows:

$$C_{11} = \left(\frac{1}{h}\right)^\alpha, \quad C_{12} \cdot \mathbf{n} = \text{sign}(\mathbf{v} \cdot \mathbf{n})/2, \quad (20)$$

where  $\mathbf{v}$  is an arbitrary but fixed vector with nonzero components.

We also define the artificial outflow boundary as

$$\partial\Omega^+ = \{(x, y) \in \partial\Omega \mid \mathbf{v} \cdot \mathbf{n} \geq 0\}. \quad (21)$$

Then the artificial inflow can be expressed as

$$\partial\Omega^- = \partial\Omega \setminus \partial\Omega^+. \quad (22)$$

The addition of the artificial outflow boundary  $\partial\Omega^+$  and artificial inflow boundary  $\partial\Omega^-$  equals the total boundary  $\partial\Omega$ . In this paper,  $\mathbf{v} = (1, 1)^t$ , and then, the east boundary and the north boundary form the artificial outflow boundary.

In numerical calculation, numerical polynomial solution  $U$  in each element is approximated using different elements as

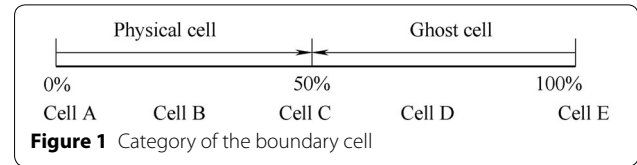
$$U_h = \sum_{i=1}^N U_i(t) B_i(x). \quad (23)$$

### 3 New Immersed Boundary Method

The numerical procedure we use is as follows:

1. Detect the boundary and determine boundary cells. Record intersection of the boundary and the cells.
2. Detect the ghost cells in the boundary cells, and record the type of the boundary.
3. Restructure the ghost cells to impose the boundary condition implicitly.
4. Solve the equations iteratively using the successive substitution method.
5. Repeat step (3) and step (4) until the convergence is achieved.

Here, the boundary cell is the cell cut by the boundary and recording the coordinates of the intersection aids preparing for the following reconstruction. All cells are divided into two categories: physical cells and ghost cells. The cell is a ghost cell when the cell center is outside the calculation domain, and when the cell center is within the



domain, it is a physical cell. The cell categories are shown in Figure 1.

The cell is a physical cell when the volume fraction out of the calculation domain ranges from 0% to 50%. Cell A is the cell whose the volume fraction equals 0%, and the edge of the cell coincide with the boundary. Therefore, cell A has the smallest error. Cell B is a cell whose the volume fraction is from 0% to 50%. Although the boundary also is imposed on the cell, the edge of the cell does not coincide with the boundary. Cells A and B belong to the physical cell category, and the other three cells are the ghost cells. Cell E is the ghost cell whose the volume fraction equals 100% and this type of cell only has one intersection with the boundary. The neighbor cell of Cell E has no physical cell, and so, cell E has no influence on the calculation. Cell C is the ghost cell that the volume fraction equals to 50% and has biggest influence on calculation domain. Therefore, cell C has the biggest error. Cell D is the ghost cell that the volume fraction is from 50% to 100%. After the above analysis, the arrows in Figure 1 denote the directions of increasing error.

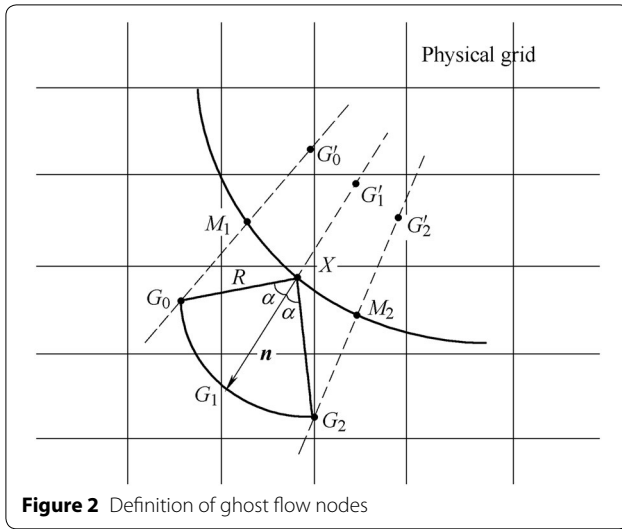
In our procedure, calculation occurs in the physical cell when updating the data and the ghost cell is only used for imposing the boundary conditions through the reconstruction. Therefore before updating the data, we must reconstruct the ghost cell.

We employ the information of the ghost node in the ghost cell to restructure the numerical polynomial solution of the LDG method. The procedure of reconstruction involves two steps: identifying the information of the ghost cell and reconstruction

#### 3.1 Identify the Information of the Ghost Cell

The method is depicted in Figure 2. To reconstruct the numerical polynomial solution in the ghost cell, we need information of boundary node  $X$  and flow node  $G$ . To avoid the numerical instability, the boundary node is chosen. Now, we identify flow node  $G$ .

We first determine the position of the ghost node. The outward normal vector on boundary node  $X$  should be specified, and then the vector is rotated  $\alpha^\circ$  clockwise and then  $\alpha^\circ$  anticlockwise. When  $\alpha^\circ$  is a different value, there are different odd directions. Then, the positions of the ghost nodes are ascertained at a distance  $R$  from boundary node  $X$  in the different directions. To minimize the



**Figure 2** Definition of ghost flow nodes

property of numerical instability, the ghost node is chosen symmetrically.

Second, we locate the image of the node inside the flow domain using the boundary and intersection of the boundary. Nodes  $G'_0, G'_1, G'_2$  are the images of ghost nodes  $G_0, G_1, G_2$ , and nodes  $M_1, X, M_2$  are the boundary intersection as shown in Figure 2. Flow variables  $\varphi'$  of nodes  $G'_0, G'_1, G'_2$  are evaluated using Eq. (3), and the value  $\varphi_0$  of the intersection ( $G'_0, G'_1, G'_2$ ) is evaluated using boundary condition. The value at the ghost node is then  $\varphi_G = 2\varphi_0 - \varphi'$ .

After the information of ghost cell and boundary node is obtained, we can start the reconstruction of the numerical polynomial solution.

### 3.2 Reconstruction

The reconstruction is decided by the shape function of the LDG method

$$U_h = \mathbf{U}\mathbf{B} \quad (24)$$

where  $\mathbf{U} = (u_0, u_1, \dots, u_N)^T$  and  $\mathbf{B} = (B_1, B_2, \dots, B_N)^T$ .

The coefficients can be expressed in terms of the nodal values:

$$\mathbf{U} = \mathbf{A}^{-1}\mathbf{U}_h, \quad \mathbf{U}_h = (U_0, U_1, \dots, U_N)^T, \quad (25)$$

where  $\mathbf{U}_h$  is the vector of three values, and for linear interpolation,  $\mathbf{A}$  is an  $N \times N$  matrix whose elements can be computed from the coordinates of the three nodes, when the velocity at the boundary is specified.

$$\mathbf{A} = \begin{bmatrix} B_1^1(x_1, y_1) & B_2^1(x_1, y_1) & \dots & B_N^1(x_1, y_1) \\ B_1^2(x_2, y_2) & B_2^2(x_2, y_2) & \dots & B_N^2(x_2, y_2) \\ \vdots & \vdots & \ddots & \vdots \\ B_1^N(x_N, y_N) & B_2^N(x_N, y_N) & \dots & B_N^N(x_N, y_N) \end{bmatrix}. \quad (26)$$

where  $(x_1, y_1), (x_2, y_2)$  and  $(x_N, y_N)$  are the coordinates of the boundary node or the ghost nodes. To minimize the property of the numerical instability, the ghost nodes should be chosen symmetrically.

### 4 Numerical Example

In the numerical experiment, the shape function is the first-order Legendre shape function and multi-dimensional polynomials are formed by “tensor-product” approximations to preserve the orthogonal property for mass matrices. All the meshes are generated by the same code of the adaptive Cartesian grid. The interested reader can refer to the books of Samet for more details.

We compute the  $L^2$ -norm error,  $L^\infty$ -norm error and the order of convergence.

The  $L^2$ -norm error measure is given by

$$\|e(x)\|_{L^2} = \|u(x) - u_h(x)\|_{L^2} = \left( \int_{\Omega} |u(x) - u_h(x)|^2 dv \right)^{\frac{1}{2}} \quad (27)$$

The  $L^\infty$ -norm error measure is given by

$$\|e(x)\|_{(L^\infty)} = \|u(x) - u_h(x)\|_{(L^\infty)} = \max |u(x) - u_h(x)|. \quad (28)$$

We consider the linear diffusion problem

$$-\Delta u = f(x, y), \quad (x, y) \in \Omega = (0, 0.32)^2, \quad (29)$$

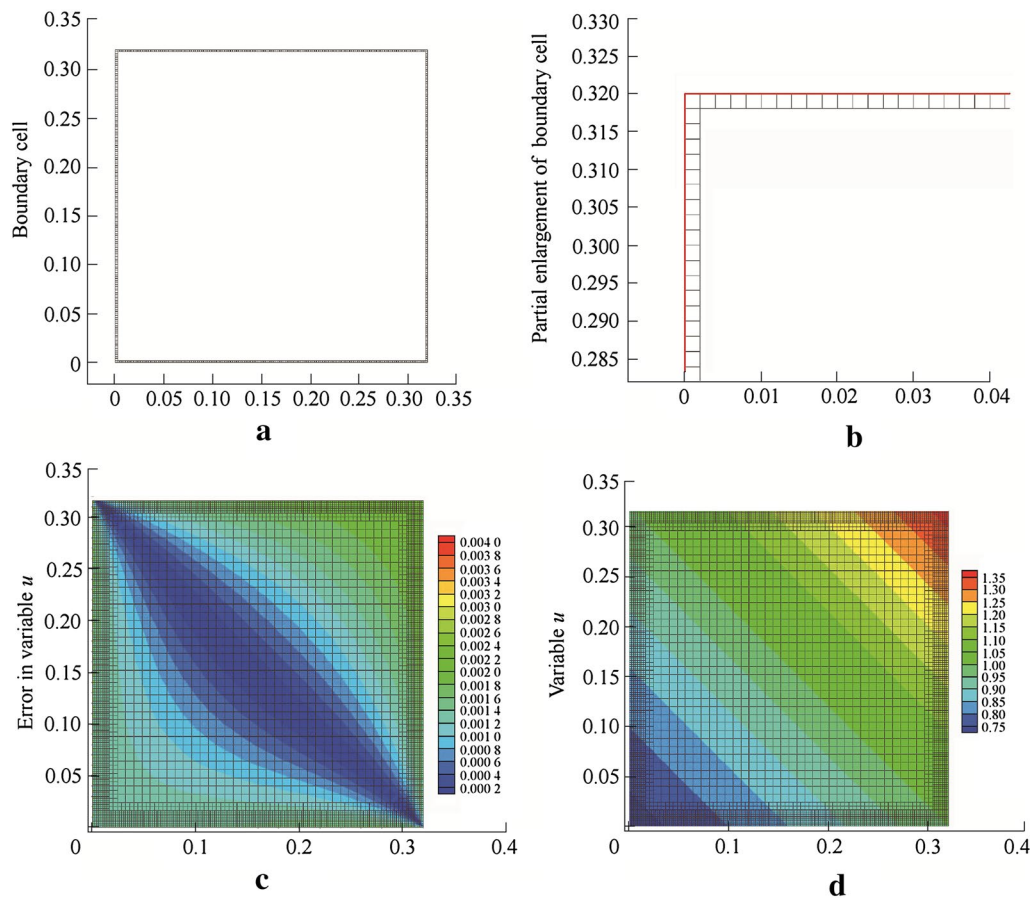
and select the boundary condition and  $f(x, y)$ , such that the true solution is

$$u(x, y) = e^{x+y}. \quad (30)$$

The gradient is defined as  $\mathbf{q} = (q_1, q_2)$ . Because of the symmetry of the problem, the orders and error of convergence are same for  $q_1$  and  $q_2$ . In the table, we show only the error and orders of  $q_1$ .  $N_B$  denotes the number of boundary cells:  $N_S$  denotes the number of starting cells,  $N_G$  denotes the number of ghost cells, and  $N_T$  denotes the number of the total cells.

#### 4.1 Boundary Condition is Imposed Through Numerical Fluxes

In this example, the boundary cell and boundary are as shown in Figures 3(a) and 3(b), and the numbers of the different types of cells are presented in Table 1. We can see that there is no ghost cell in the boundary cell and cell A is only the ghost cell. Figures 3(c) and 3(d) show the distribution of the error and  $u$ . There is the maximum error in the artificial outflow boundary. The type of the boundary cell is presented in Table 1. There is no ghost cell and all boundary cells are cell A. It does not need reconstruction and the boundary condition is imposed through numerical fluxes.



**Figure 3** Boundary cell and error distribution by imposing the boundary condition through numerical fluxes: (a) Boundary cell; (b) Partial enlargement of the boundary cell with the boundary; (c) Error distribution of variable  $u$ ; (d) Distribution of variable  $u$

**Table 1** Numbers of cells with different starting cells

$i$	$N_s$	$N_B$	$N_G$	$N_T$
1	$20 \times 20$	316	0	2740
2	$30 \times 30$	476	0	4560
3	$40 \times 40$	636	0	6580
4	$50 \times 50$	796	0	8800
5	$60 \times 60$	950	0	11220
6	$80 \times 80$	1276	0	16660

The error and convergence rate are presented in Table 2. We observe that  $L_\infty$  and  $L_2$  in  $q$  achieve 3.5th and 2.0th order super convergence and that  $L_\infty$  and  $L_2$  in  $u$  all achieve 1.5th order super convergence.

#### 4.2 Boundary Condition is Imposed Through Reconstruction

In this example, Figures 4(a) and 4(b) show the boundary cell and boundary. The boundary cell includes three

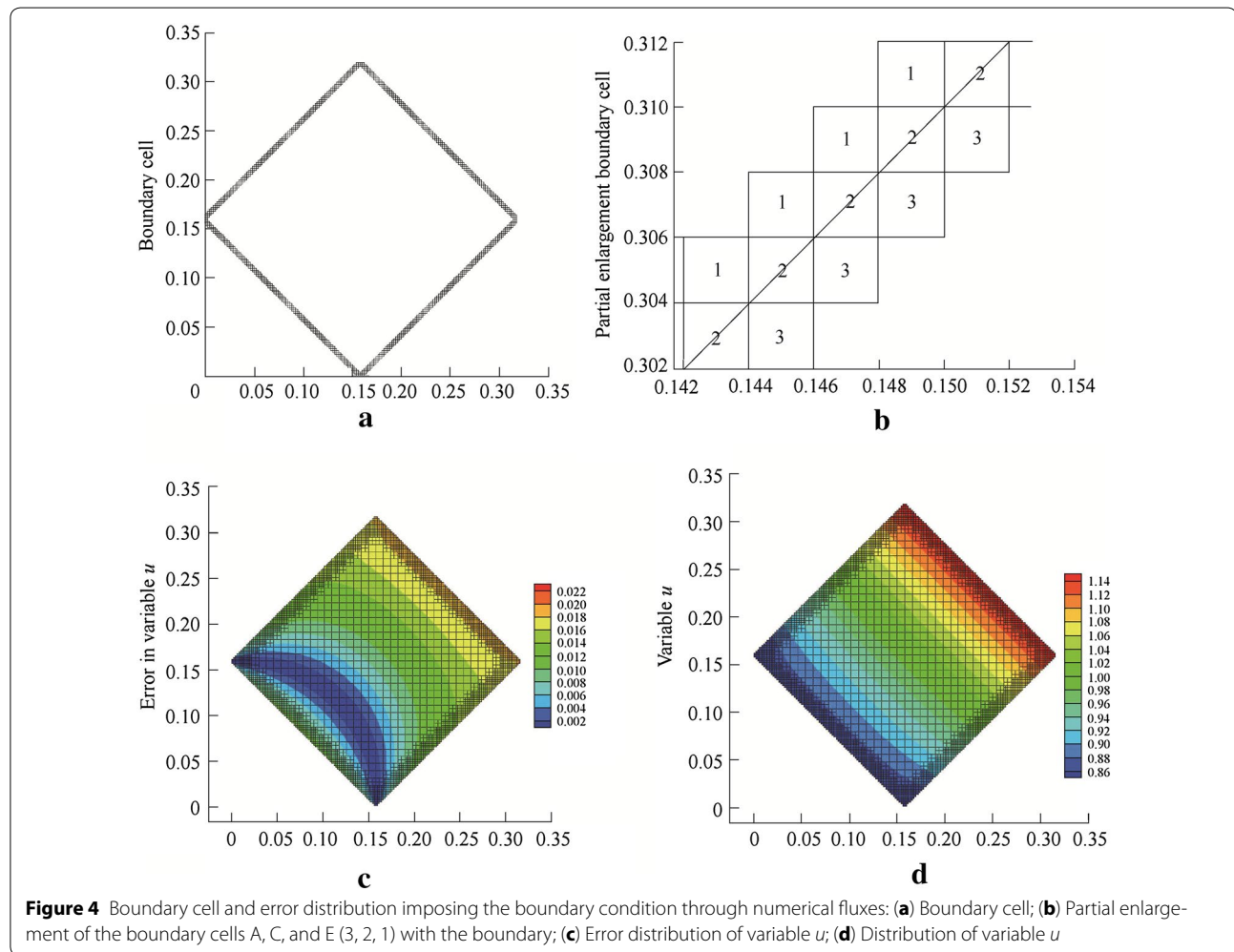
categories: Cells A, C and E. Cells A and E are the ghost cells and Cell C is the physical cell. Cells A and E have no influence on imposing the boundary condition. The boundary condition is imposed through reconstruction in Cell C. The numbers of the different type cell is listed in Table 3. There are more boundary cells and total cells than those in Section 4.1 and the boundary is more complicated. From Figure 4(c), we can see that the maximum error is in the artificial outflow boundary. From Figure 4(d), we can see that there is a smooth curve in distribution in  $u$  instead of linear distribution described in Section 4.1. This shows that the error is larger.

The error and order of convergence are listed in Table 4, and we observe that the convergence order of  $L^\infty$  and  $L^2$  in  $u$  is the 1.5th order, and it is equal to that described in the Section 4.1. The convergence order of  $L_\infty$  and  $L_2$  in  $q$  reach 1.5th and 2.0th order super convergence and the values are lower than the case described in Section 4.1.



**Table 2** Errors and orders of convergence on different cells

<i>i</i>	$q_1$				$u$			
	$L^\infty$	Order	$L^2$	Order	$L^\infty$	Order	$L^2$	Order
1	$2.55 \times 10^{-5}$		$8.79 \times 10^{-5}$		$4.63 \times 10^{-3}$		$7.17 \times 10^{-4}$	
2	$1.05 \times 10^{-5}$	3.48	$4.72 \times 10^{-5}$	2.44	$3.10 \times 10^{-3}$	1.58	$4.81 \times 10^{-4}$	1.57
3	$5.51 \times 10^{-6}$	3.51	$3.05 \times 10^{-5}$	2.39	$2.32 \times 10^{-3}$	1.56	$3.63 \times 10^{-4}$	1.54
4	$3.30 \times 10^{-6}$	3.53	$2.17 \times 10^{-5}$	2.33	$1.87 \times 10^{-3}$	1.53	$2.91 \times 10^{-4}$	1.51
5	$2.14 \times 10^{-6}$	3.57	$1.65 \times 10^{-5}$	2.28	$1.56 \times 10^{-3}$	1.49	$2.43 \times 10^{-4}$	1.48
6	$1.04 \times 10^{-6}$	3.66	$1.07 \times 10^{-5}$	2.20	$1.17 \times 10^{-3}$	1.45	$1.83 \times 10^{-4}$	1.44



### 4.3 Boundary Condition is Imposed Hybrid

In this example, the boundary is a curve, and the category of the boundary cell is generated automatically according to the generation algorithm of the adaptive Cartesian grid. It tests the error and super convergence at the arbitrary boundary and here we hope to find the impact factors of error change. The manner of imposing the boundary condition depends on the type of the boundary.

**Table 3** Numbers of cells with different starting cells

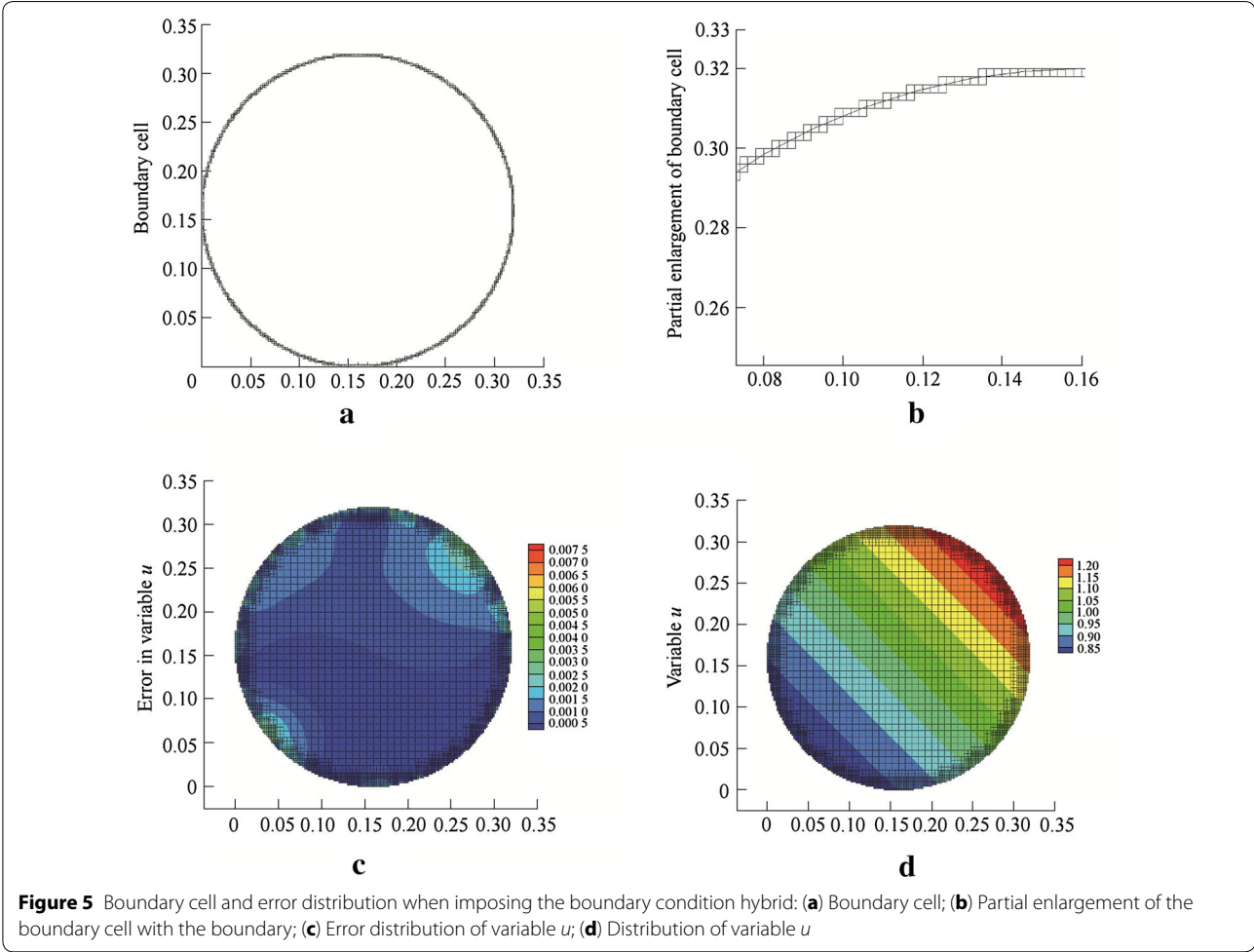
<i>i</i>	$N_s$	$N_B$	$N_G$	$N_T$
1	$20 \times 20$	472	316	1804
2	$30 \times 30$	712	476	3020
3	$40 \times 40$	952	636	4300
4	$50 \times 50$	1192	796	5680
5	$60 \times 60$	1432	956	7160
6	$80 \times 80$	1912	1276	10420

Table 4 Errors and orders of convergence on different cells

<i>i</i>	<i>q</i> <sub>1</sub>				<i>u</i>			
	<i>L</i> <sup>∞</sup>	Order	<i>L</i> <sup>2</sup>	Order	<i>L</i> <sup>∞</sup>	Order	<i>L</i> <sup>2</sup>	Order
1	1.59 × 10 <sup>−4</sup>		2.21 × 10 <sup>−4</sup>		3.85 × 10 <sup>−2</sup>		5.10 × 10 <sup>−3</sup>	
2	9.94 × 10 <sup>−5</sup>	1.84	1.21 × 10 <sup>−4</sup>	2.35	2.63 × 10 <sup>−2</sup>	1.49	3.48 × 10 <sup>−3</sup>	1.50
3	7.23 × 10 <sup>−5</sup>	1.75	7.89 × 10 <sup>−5</sup>	2.35	2.00 × 10 <sup>−2</sup>	1.50	2.64 × 10 <sup>−3</sup>	1.51
4	5.68 × 10 <sup>−5</sup>	1.68	5.66 × 10 <sup>−5</sup>	2.31	1.61 × 10 <sup>−2</sup>	1.49	2.13 × 10 <sup>−3</sup>	1.50
5	4.68 × 10 <sup>−5</sup>	1.62	4.31 × 10 <sup>−5</sup>	2.28	1.35 × 10 <sup>−2</sup>	1.47	1.78 × 10 <sup>−3</sup>	1.47
6	3.46 × 10 <sup>−5</sup>	1.56	2.80 × 10 <sup>−5</sup>	2.22	1.02 × 10 <sup>−2</sup>	1.44	1.35 × 10 <sup>−3</sup>	1.44

A circular boundary is chosen because the slope varies from  $-\infty$  to  $+\infty$ . Magnified views are shown in Figures 5(a) and 5(b). There are all types of boundary cells, and it difficult to distinguish every cell. The number of cells is listed in Table 5. The numbers of the boundary cell, ghost cell and total cell in this example lie between the corresponding numbers for the examples in Sections 4.1 and 4.2.

In Figure 5(c), there is the maximum error in boundary, but it is more complicated than in the previous two examples. From Figure 5(d), it is clear that the distribution in  $u$  is better than that in Section 4.2. The error and convergence order are listed in Tables 6 and 7. It is obvious than the error lies between the error in the two example above. Good convergence rate is achieved. The error decreases with the increase of grid



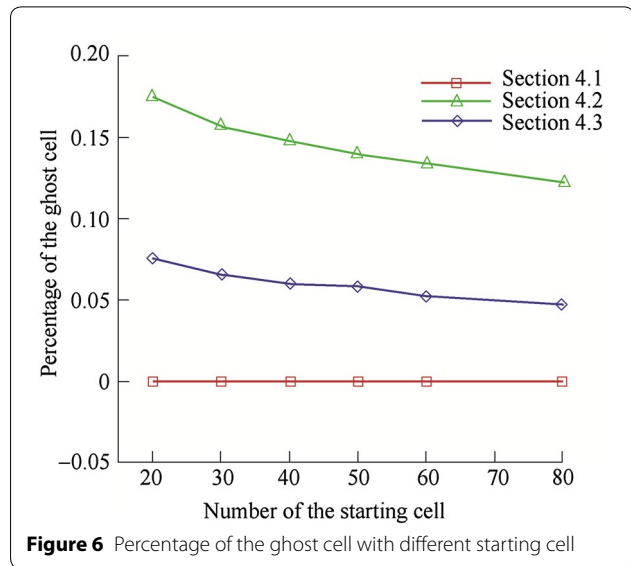
**Table 5** Numbers of the cells for different starting cell

$i$	$N_s$	$N_B$	$N_G$	$N_T$
1	$20 \times 20$	324	156	2072
2	$30 \times 30$	484	220	3352
3	$40 \times 40$	644	288	4808
4	$50 \times 50$	811	383	6587
5	$60 \times 60$	964	456	8760
6	$80 \times 80$	1284	572	12124

number, but it is abnormal that the order of the convergence oscillates. We need to analyze the reason for this oscillation.

In Figure 4(b), the boundary cell includes a variety of cells, and it is obvious that Cells B and D form the large majority. According to the algorithm of the adaptive Cartesian grid, the number of every cell is out of control. If the total number of the cell is increasing, all physical cells infinitely approach Cell A, and the error of all ghost cell must be lower than that of cell.

The percentage of ghost cells decreases with an increase in the starting cell as shown in Figure 6, and the error decreases with increase in the total number of cells as shown in Figure 7. It is clear that the error in the

**Figure 6** Percentage of the ghost cell with different starting cell

complex geometries is close to that in the case that the grid comprise only all physical cells.

The conclusion is derived that the algorithm works well in complex geometries and that the change in the percentage of different cells is the main cause of the oscillation of the order of convergence.

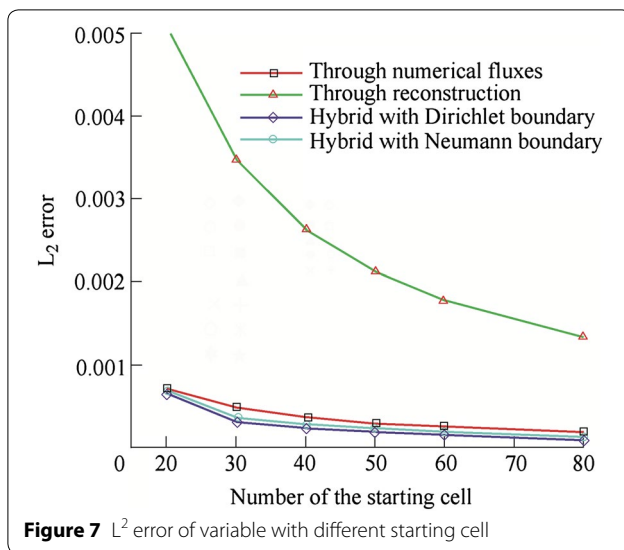
**Table 6** Errors and orders of convergence on the different cell with the Dirichlet boundary

$i$	$q_1$				$u$			
	$L^\infty$	Order	$L^2$	Order	$L^\infty$	Order	$L^2$	Order
1	$1.05 \times 10^{-4}$		$1.27 \times 10^{-4}$		$8.39 \times 10^{-3}$		$6.75 \times 10^{-4}$	
2	$3.63 \times 10^{-4}$	-5.05	$6.83 \times 10^{-5}$	2.52	$5.31 \times 10^{-3}$	1.86	$3.34 \times 10^{-4}$	2.87
3	$8.76 \times 10^{-5}$	7.48	$4.59 \times 10^{-5}$	2.09	$4.93 \times 10^{-3}$	0.39	$2.61 \times 10^{-4}$	1.29
4	$4.82 \times 10^{-5}$	3.94	$3.27 \times 10^{-5}$	2.22	$5.21 \times 10^{-3}$	-0.36	$2.18 \times 10^{-4}$	1.20
5	$1.22 \times 10^{-5}$	9.44	$2.51 \times 10^{-5}$	1.83	$4.85 \times 10^{-3}$	0.48	$1.71 \times 10^{-4}$	1.67
6	$1.22 \times 10^{-5}$	0.00	$1.62 \times 10^{-5}$	2.65	$3.15 \times 10^{-3}$	2.62	$1.02 \times 10^{-4}$	3.15

**Table 7** Errors and orders of convergence on the different cell with the Neumann boundary

$i$	$q_1$				$u$			
	$L^\infty$	Order	$L^2$	Order	$L^\infty$	Order	$L^2$	Order
1	$1.25 \times 10^{-4}$		$1.25 \times 10^{-4}$		$8.65 \times 10^{-3}$		$7.06 \times 10^{-4}$	
2	$3.59 \times 10^{-4}$	-4.33	$6.72 \times 10^{-5}$	2.48	$5.32 \times 10^{-3}$	1.99	$3.56 \times 10^{-4}$	3.05
3	$5.80 \times 10^{-5}$	9.60	$4.51 \times 10^{-5}$	2.10	$5.22 \times 10^{-3}$	0.10	$2.83 \times 10^{-4}$	1.20
4	$4.51 \times 10^{-5}$	1.66	$3.22 \times 10^{-5}$	2.23	$5.70 \times 10^{-3}$	-0.57	$2.33 \times 10^{-4}$	1.26
5	$2.92 \times 10^{-5}$	2.99	$2.48 \times 10^{-5}$	1.79	$5.30 \times 10^{-3}$	0.50	$1.86 \times 10^{-4}$	1.58
6	$2.77 \times 10^{-7}$	28.21	$1.59 \times 10^{-5}$	2.69	$3.23 \times 10^{-3}$	3.00	$1.11 \times 10^{-4}$	3.10





## 5 Conclusions

1. The traditional IBM applied to the numerical construction of discrete points is developed for the new IBM applied to the reconstruction of the distribution function in LDG method.
2. The numerical performance of the LDG method is studied in the grid with a hanging node (the adaptive Cartesian grid). The numerical result shows that the method works well.
3. The boundary condition is imposed through a combination of the numerical flow and reconstruction of distribution function in LDG method. When the boundary condition is imposed through numerical fluxes, the error is minimum. When the boundary condition is imposed through reconstruction, the error is maximum.
4. From error distribution, the maximum error of the variable always exists at the boundary. Neither reconstruction nor numerical fluxes is directly imposed on the condition in a finite volume method or a finite element method. This is the main reason for the difference in error distribution compared with the classical numerical methods.
5. The presented method works well in complex geometries. The error and convergence rate are adopted for evaluating the accuracy. The error depends on the number and type of the cell and is always near the physical cell.

### Authors' Contributions

Y-YZ and X-JZ was in charge of the whole trial; X-JZ wrote the manuscript; X-JZ, Y-SZ and KY assisted with sampling and laboratory analyses. All authors read and approved the final manuscript.

### Authors' Information

Xu-Jiu Zhang, born in 1980, is currently a PhD candidate at Key Laboratory of Education Ministry for Modern Design and Rotor-bearing System, Xi'an Jiaotong University, China. His research interest is mechanical calculation of the fluid. E-mail: zhangxuj@stu.xjtu.edu.cn.

Yong-Sheng Zhu, born in 1973, is currently a professor at Key Laboratory of Education Ministry for Modern Design and Rotor-bearing System, Xi'an Jiaotong University, China. His main research interests include bearing, rotor bearing system theoretical calculation and operation of the status of monitoring and other aspects of the study. E-mail: yszhu@mail.xjtu.edu.cn.

Ke Yan, born in 1984, is currently associate professor at Key Laboratory of Education Ministry for Modern Design and Rotor-bearing System, Xi'an Jiaotong University, China. His main research interests include the rolling bearing and rotor shaft thermal characteristic design, lubrication mechanism and the realization technology. E-mail: yanke@mail.xjtu.edu.cn.

You-Yun Zhang, born in 1947, is currently an associate professor at Key Laboratory of Education Ministry for Modern Design and Rotor-bearing System, Xi'an Jiaotong University, China. Her main research interests include equipment condition monitoring and fault diagnosis and bearing rotor dynamics. E-mail: yyzhang1@mail.xjtu.edu.cn.

### Competing Interests

The authors declare no competing financial interests.

### Ethics Approval and Consent to Participate

Not applicable.

### Funding

Supported by National Natural Science Foundation of China (Grant No. 51405375) and National Key Basic Research and Development Program of China (973 Program, Grant No. 2011CB706606).

### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 7 April 2017 Accepted: 9 March 2018

Published online: 09 April 2018

### References

1. T Kajishima, K Taira. Immersed boundary methods. *Computational Fluid Dynamics*, Springer International Publishing, 2017.
2. E A Fadlun, R Verzicco, P Orlandi, et al. Combined immersed –boundary finite difference methods for three–dimensional complex flow simulation. *Journal of Computational Physics*, 2000, 161(1): 35–60.
3. Y H Tseng, J H Ferziger. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics*, 2003, 192(2): 593–623.
4. R Verzicco, M Fatica, G Iaccarino, et al. Flow in an impeller-stirred tank using an immersed-boundary method. *Aiche Journal*, 2004, 50(6): 1109–1118.
5. R Mittal, H Dong, M Bozkurtas, et al. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *Journal of Computational Physics*, 2008, 227(10): 4825–4852.
6. T Gao, Y H Tseng, X Y Lu. An improved hybrid Cartesian/immersed boundary method for fluid–solid flows. *International Journal for Numerical Methods in Fluids*, 2007, 55(12): 1189–1211.
7. A F Shinn, M A Goodwin, S P Vanka. Immersed boundary computations of shear- and buoyancy-driven flows in complex enclosures. *International Journal of Heat and Mass Transfer*, 2009, 52(17–18): 4082–4089.
8. J Lee, D You. An implicit ghost-cell immersed boundary method for simulations of moving body problems with control of spurious force oscillations. *Journal of Computational Physics*, 2013, 233: 295–314.
9. O Hu, N Zhao, J M Liu. A ghost cell method for turbulent compressible viscous flows on adaptive Cartesian grids. *Procedia Engineering*, 2013, 67: 241–249.

10. J X Jun, K Luo, J R Fan. A ghost-cell based high-order immersed boundary method for inter-phase heat transfer simulation. *International Journal of Heat and Mass Transfer*, 2014, 75: 302-312.
11. K Luo, C Mao, Z Zhuang, et al. A ghost-cell immersed boundary method for the simulations of heat transfer incompressible flows under different boundary conditions Part-II: Complex geometries. *International Journal of Heat & Mass Transfer*, 2017, 104: 98-111.
12. K Luo, Z Zhuang, J Fan, et al. A ghost-cell immersed boundary method for simulations of heat transfer in compressible flows under different boundary conditions. *International Journal of Heat & Mass Transfer*, 2016, 92: 708-717.
13. M Wang, Y T Feng, G N Pande, et al. Numerical modelling of fluid-induced soil erosion in granular filters using a coupled bonded particle lattice Boltzmann method. *Computers & Geotechnics*, 2017, 82: 134-143.
14. R Y Li, C M Xie, Huang W X, et al. An efficient immersed boundary projection method for flow over complex/moving boundaries. *Computers & Fluids*, 2016, 140: 122-135.
15. C Zhu, H Luo, G Li. High-order immersed-boundary method for incompressible flows. *AIAA Journal*, 2016: 1-8.
16. M Kumar, S Roy. Immersed boundary method simulation of natural convection over fixed and oscillating cylinders in square enclosure. *International Journal of Heat & Fluid Flow*, 2016, 61: 407-424.
17. B Cockburn, G Kanschat, I Perugia, et al. Super convergence of the local discontinuous Galerkin method for elliptic problems on Cartesian grids. *SIAM J. NUMER. ANAL.*, 1998, 39(1): 264-285.
18. S Adjerid, M Baccouch. A superconvergent local discontinuous Galerkin method for elliptic problems. *J. Sci. Comput.*, 2012, 52: 113-152.
19. M Baccouch, S Adjerid. A posteriori LDG error estimation for two dimensional convection - diffusion problems. *Journal of Scientific Computing*, 2015, 62(2): 399-430.
20. Z Zhang, Z Xie, Z Zhang. Super convergence of discontinuous Galerkin methods for convection-diffusion problems. *Journal of Scientific Computing*, 2009, 41(1): 70-93.
21. T P Fries, A Byfut A Alizada. Hanging nodes and XFEM. *International Journal for Numerical Methods in Engineering*, 2011, 86(4-5): 404-430.
22. A R Owens, J A Welch J Kópházi. Discontinuous isogeometric analysis methods for the first-order form of the neutron transport equation with discrete ordinate (SN) angular discretisation. *Computer Methods in Applied Mechanics & Engineering*, 2016, 315: 501-535.
23. Q Liang. A simplified adaptive Cartesian grid system for solving the 2D shallow water equations. *International Journal for Numerical Methods in Fluids*, 2012, 69(2): 442-458.
24. D G Dommermuth, M Sussman, R F Beck, et al. The numerical simulation of ship waves using Cartesian grid methods with adaptive mesh refinement. *Physics*, 2014, 18(5): 597-605.
25. C Min, F Gibou. A second order accurate projection method for the incompressible Navier-Stokes equations on non-graded adaptive grids. *J. Compute. Phys.*, 2006, 219: 912-929.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)