High Order Accurate Curved-element Implementation in OpenFOAM for Discontinuous Galerkin Method

Li-Yang Xu,1, Shuai Ye,1, Yong-Quan Feng,1, Hao Li,1, Xin-Hai Xu,1, Xiao-Guang Ren,1,*

1 State key Laboratory of High Performance Computing National University of Defense Technology College of Computer National University of Defense Technology Changsha, China
{xuliyang08, yeshuai09, fengyongquan12, lihao, xuxinhai, renxiaoguang}@nudt.edu.cn

Abstract. A high order accurate discontinuous galerkin method based curved-element implementation in OpenFOAM is presented in this work. The degree of freedom points on origin straight side mesh are mapped to curvilinear edges by finding nearest points on boundary. Then the deformation are transported to the inner points by blended method. Based on the run-time dynamic code of OpenFOAM, we design a convenient interface for users to configure the parameterized function of curvilinear boundary. The curved-element implementation is applied to the simulation of vacuum Maxwell equations in metallic air-filled cavity. Results show that the curved-element has significantly improved the mesh quality. High order numerical convergence rate has been achieved.

Keywords: curved-element, high order, discontinuous galerkin, OpenFOAM

1 Introduction

In the last few decades, computational fluid dynamics (CFD) has been more and more important in both scientific research[1]2] and industrial manufacture. At the meantime, significant progress has been made in the research of high order numerical methods[3]. Comparing to low order methods, high order methods can provide more accurate simulation results in the same computational costs. Discontinuous Galerkin Method (DGM) is one of the most popular and widely used high order method. DGM was firstly proposed by Reed and Hill in the 1970s to solve the neutron transport problem[4]. Cockburn and Shu introduced Runge-Kutta time stepping to DGM and succeed in simulation of conservation laws system[5]–[7]. Based on their creative efforts, a serial of work has extended the DGM to most of the CFD area.

* State Key Laboratory of High Performance Computing, National University of Defense Technology, Changsha, Hunan, China. Email: renxiaoguang@nudt.edu.cn

ISSN: 2287-1233 ASTL Copyright © 2016 SERSC
The released DGM software, such as Deal.II [8] and Fenics [9], have complex user interface and weak support for curved boundary. Better designed DGM software is an urgent task for CFD. OpenFOAM[10] is a mature open source CFD software based on finite volume method. Due to its excellent frame work design and convenient interfaces, it gained great reputation in CFD community. However, the numerical discretization of OpenFOAM is 2nd order at most. Our group is trying to introduce DGM in OpenFOAM to realize high order numerical simulation, and provide convenient user interface consistent to OpenFOAM. The main work of this paper include:

1. Implementation of high order accurate curved-element in OpenFOAM for DGM;
2. Design of convenient interface to configure parameterized curve function based on run-time dynamic code;
3. Simulation of Maxwell equations in a metallic air-filled cavity to validate the curved-element and convergence rate.

The rest of the paper is organized as follows. Section 2 briefly introduce the main feature of DGM and the representation of curved boundary. The algorithm to calculate the curved-element and software design are described in section 3. Section 4 demonstrate the effects of curved element in geometry and numerical accuracy with the simulation of Maxwell equation. Conclusions are in Section 5.

2 Discontinuous Galerkin Method

We adopt the nodal DGM originally proposed by Hesthaven and Warburton[11]. Here we briefly introduce the numerical scheme. Taking $u$ as the unknowns, Eq. (1) can be rewritten as conservation law in Eq. (2).

$$\partial_t u + \nabla \cdot F(u) = 0, \quad x \in \Omega, \quad t > 0,$$

$$u = u^0, \quad t = 0. \tag{2}$$

The computational domain $\Omega$ is divided into a collection of non-overlapping elements.

$$\Omega = \bigcup_{i=1}^N \Omega_i. \tag{3}$$

Then, we multiply the conservation law by a test function $\varphi$ and integrate over each element $\Omega_i$. By using the green theory, we finally get

$$\int_{\Omega_i} \varphi \partial_t u ds + \int_{\partial \Omega_i} \varphi F(u) \cdot n d\tau - \int_{\Omega_i} \nabla \cdot F(u) ds = 0. \tag{4}$$

Where $n$ represents the normal vector on $\partial \Omega_i$. The unknowns $U$ can be described as the linear combination of nodal Lagrange basis $\{\ell_i\}_{i=1}^N$ or modal basis $\{\varphi_i\}_{i=1}^N$, where
\[ u_k = \sum_{i=0}^{N_u} u_{i,j} \phi_i(x) = \sum_{j=0}^{N_p} c_{i,j} \phi_{i,j}(r), k = 1, 2, 3, 4, \] (5)

3 Methodology

This section firstly introduces the algorithm to calculate the DOF location of curved-element. To provide a convenient interface for users to describe and configure the curved boundary, we design a dictionary-based interface in the second part.

Consider a curved triangle element in domain \( \Omega \) with boundary \( \Gamma \) defined by parameterized function \( S(a,b) \). In most CFD software, edges in mesh are all straight as shown in the left subfigure of Fig. 1. There is a gap between the straight edges with the analytical curvilinear edge (the dotted line). The straight sided mesh will introduce error to the DGM discretization. To achieve consistent high order accuracy, the DOF locations should be redistributed according to the curved boundary.

![Fig. 1. Left: Original straight side element; Middle: One face made curvilinear; Right: Deformation Blended into inner points.](image)

Firstly, we focus on the DOF points on the boundary face and trying to map them to the curve \( S(a,b) \). The deformation of DOF points on the face can be found by solving

\[ \min_{(a,b)} \frac{1}{2} \| x_M - S(a,b) \|^2, \] (7)

where \( x_M \) is the DOF location on the origin straight side mesh. We use Newton-Gauss iteration to solve this nonlinear problem, and obtain the mapped points \( x_S \) on the curve. With the mapped points, the curve now can be represented by the basis in \( n \)th order form analytically as Eqn. (8). The difference \( (\Delta x) \) between \( x_M \) and \( x_S \) represents the deformation of facial DOF points.

\[ \tilde{S}(x) = \sum_{j=0}^{N_r} x_{j,i} \phi_{j} (x). \] (8)
Secondly, we focus on the inner DOF points in the curved-element. The inner DOF should be shifted consisted with facial DOF points. The deformation is transported to inner points by blended rule to remain the LGL property.

The method introduced in the previous subsection can deal with any kind of curve, but the parameterized function $S(a,b)$ should be provided as the precondition. A simple way is to write the function in code directly, which is inflexible and the code has to be recompiled if the curve changed. OpenFOAM includes the capability to compile, load and execute C++ code at run-time. The component called codeStream can be used to generate user defined interface from configuration dictionary. Inheriting from this component, we implement an interface to provide a swift way to configure custom curvilinear function.

4 Experiments and Results

To demonstrate the effects of curved-element, we simulate the vacuum Maxwell equations in a metallic air-filled cavity. A series of meshes with different size are used to test the order of convergence. Boundary of the computational domain is a circle with radius 1.0. Three meshes with size $h$, $h/2$ and $h/4$ are generated by Gambit. Curved boundaries are approximated by piecewise straight edges.

DGM curved elements mainly take effects on two aspects, the geometry of boundary elements and numerical convergence rate. After applied curved element, the DOF points in the outermost elements are shifted according to the parameterized curvilinear function. Fig. 2 illustrate the details of mesh deformation. It is clear that the curved elements have significantly improved the mesh quality. The DOF points are located almost exactly on the analytical curvilinear boundary.

![Fig. 2. Deformation of curved element on boundary faces. Left: original straight side mesh; Right: mesh with 4th order curved-elements.](image)

We run the case with a transient solver based on OpenFOAM and DGM. Simulation results obtained from mesh with size $h/2$ and 4th order at time $= 0.5$ is showed in Fig. 3.
Fig. 3. Simulation result of electric field $E$ with 4th order curved-elements at time=0.5, on the mesh with size $h/2$.

To check the convergence order, we run the case in a sequence of meshes and different base orders ranging from 1 to 6. Table 1 lists the $L^2$ error and convergence rate simulated by DGM without curved-elements, while Table 2 lists the results by DGM with curved-elements. Though the order of basis increasing from 1 to 6, the convergence rate remains 2. Contrast to this, Table 2 shows that simulation with curved elements can reach the theoretical convergence rate.

Table 1. Results without curved-elements. $L^2$ error and convergence rate of electric field $E$ obtained from the simulation based on different basis order and mesh size. The $h/4$ mesh can not get converged when $N$ equals 5 or 6.

<table>
<thead>
<tr>
<th>N</th>
<th>$h$</th>
<th>$h/2$</th>
<th>$h/4$</th>
<th>Convergence rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.33e-01</td>
<td>1.82e-01</td>
<td>3.68e-02</td>
<td>1.59</td>
</tr>
<tr>
<td>2</td>
<td>1.82e-01</td>
<td>5.09e-02</td>
<td>8.00e-03</td>
<td>2.25</td>
</tr>
<tr>
<td>3</td>
<td>1.19e-01</td>
<td>2.15e-02</td>
<td>1.15e-02</td>
<td>2.69</td>
</tr>
<tr>
<td>4</td>
<td>7.18e-02</td>
<td>1.15e-02</td>
<td>1.15e-02</td>
<td>2.30</td>
</tr>
<tr>
<td>5</td>
<td>4.79e-02</td>
<td>1.11e-02</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>6</td>
<td>4.87e-02</td>
<td>1.15e-02</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

Table 2. Results with curved-elements. $L^2$ error and convergence rate of electric field $E$ obtained from the simulation based on different basis order and mesh size.

<table>
<thead>
<tr>
<th>N</th>
<th>$h$</th>
<th>$h/2$</th>
<th>$h/4$</th>
<th>Convergence rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.33e-01</td>
<td>1.82e-01</td>
<td>3.68e-02</td>
<td>1.59</td>
</tr>
<tr>
<td>2</td>
<td>1.70e-01</td>
<td>4.79e-02</td>
<td>7.72e-03</td>
<td>2.23</td>
</tr>
<tr>
<td>3</td>
<td>8.72e-02</td>
<td>1.70e-02</td>
<td>8.04e-04</td>
<td>3.38</td>
</tr>
<tr>
<td>4</td>
<td>4.10e-02</td>
<td>2.44e-03</td>
<td>7.09e-05</td>
<td>4.59</td>
</tr>
<tr>
<td>5</td>
<td>9.67e-03</td>
<td>4.68e-04</td>
<td>4.95e-06</td>
<td>5.47</td>
</tr>
<tr>
<td>6</td>
<td>3.02e-03</td>
<td>4.75e-05</td>
<td>3.30e-07</td>
<td>6.58</td>
</tr>
</tbody>
</table>
5 Conclusion

A high order accurate DGM based curved-element has been implemented in OpenFOAM. The curved-element has significantly improved the mesh quality for curvilinear boundary. High order accurate convergence can be achieved in DGM simulation. Describing the curvilinear boundary function is simple and direct. With the support of run-time dynamic code stream, parameterized functions are set in the boundary control dictionary.

Though the case in experiment is simple, this implementation can be used for much more complex cases. Extending to 3-dimension case is natural. No need to make any change, this method can be applied to parallel computing. But the decomposing of curvilinear boundary should be treated specially.

Acknowledgments. The authors declare that there is no conflict of interest regarding the publication of this paper. The authors would like to thank the National Natural Science Foundation of China (Grant no. 61303071) and the Open fund from State Key Laboratory of High Performance Computing (no. 20150301 and 20150302) for funding.

References