TAIWANESE JOURNAL OF MATHEMATICS
Vol. 20, No. 5, pp. 1185–1202, October 2016
DOI: 10.11650/tjm.20.2016.7612
This paper is available online at http://journal.tms.org.tw

High-order Discontinuous Galerkin Method for Solving Elliptic Interface Problems

Min-Hung Chen* and Rong-Jhao Wu

Abstract. In this study, we develop a high-order accurate discontinuous Galerkin scheme using curvilinear quadrilateral elements for solving elliptic interface problems. To maintain accuracy for curvilinear quadrilateral elements with Q^k-polynomial basis functions, we select Legendre-Gauss-Lobatto quadrature rules with $k+2$ integration points, including end points, on each edge. Numerical experiments show quadrature rules are appropriate and the numerical solution converges with the order $k+1$. Moreover, we implement the Uzawa method to rewrite the original linear system into smaller linear systems. The resulting method is three times faster than the original system. We also find that high-order methods are more efficient than low-order methods. Based on this result, we conclude that under the condition of limited computational resources, the best approach for achieving optimal accuracy is to solve a problem using the coarsest mesh and local spaces with the highest degree of polynomials.

1. Introduction

In this study, we develop a high-order numerical scheme to solve elliptic interface problems. Numerical methods for solving elliptic interface problems are very useful in various fields, including fluid dynamics, solid mechanics, electrodynamics, material science and biological fluid mechanics. In these applications, discontinuities may arise in a solution or its derivatives across the interface.

This study examines elliptic interface problems in which the media property, the solution u, and its derivatives ∇u, have jumps across the interface. In general, there are two approaches to deal with these problems: Boundary fitted methods and immersed boundary methods.

To treat the irregular geometry of the boundary and interface, boundary fitted methods construct a computational mesh that coincides with the problem domain, e.g., the finite element methods and finite volume methods formulated on curvilinear or unstructured

Received April 9, 2016; Accepted May 12, 2016.

Communicated by Ming-Chih Lai.

2010 *Mathematics Subject Classification.* 65M60, 65N30.

Key words and phrases. High-order method, Discontinuous Galerkin method, Elliptic interface problems, Curved elements.

*Corresponding author.

meshes [1,5,14]. One big advantage of boundary fitted methods is that the boundary and interface conditions can be imposed at the boundary nodes, edges or faces. However, a drawback of boundary fitted methods is that mesh generation is difficult and sometimes time-consuming for complex geometries.

Another approach to treating irregular geometry of the boundary and interface is to generate the computational mesh encompassing the problem without following the interfaces or boundaries. Examples of this approach include the Immersed Interface Method [16], Coupling Interface Method [3], Virtual Node Method [15] and the Ghost Fluid Method [17,18]. Mesh generation is facile using this approach, but the imposition of boundary and interface conditions are very complicated.

In this study, we consider a high-order accurate discontinuous Galerkin (DG) method using curved quadrilateral elements.

In 1973, Reed and Hill proposed the first DG method to solve hyperbolic equations. Cockburn, Shu and their collaborators introduced the Runge-Kutta discontinuous Galerkin (RKDG) method for numerical solutions of hyperbolic conservation laws [6–10]. In 1998, Cockburn and Shu [11] introduced the local discontinuous Galerkin (LDG) space discretization method for time-dependent convection-diffusion systems by generalizing the RKDG method for nonlinear hyperbolic systems. The LDG method for solving elliptic problems was introduced in [4,12]. Guyomarc'h, Lee and Jeon [14] modified the LDG trace in [11] to solve elliptic interface problems. However, Guyomarc'h et al. only used straight-edge elements and could not provide high-order approximation for problems with a curved boundary or interface.

In this study, we use the modified LDG flux proposed in [14] to solve model problems. For improved accuracy, we use quadrilateral elements instead of triangular ones. In addition, we use Q^k-polynomial basis functions instead of P^k-polynomial basis functions. To better treat the curved interface and boundary, we use curvilinear quadrilateral elements that are fitted to the interface.

2. Model problem

Given a convex bounded domain $\Omega \subset \mathbb{R}^2$ and a domain Ω_1 with $\overline{\Omega}_1 \subset \Omega$ and assuming that the boundaries $\partial\Omega$ and $\partial\Omega_1$ are Lipschitz continuous, we set $\Gamma_I = \partial\Omega_1$ and $\Omega_2 = \Omega \setminus \overline{\Omega}_1$. The domains Ω_1, Ω_2 and curve Γ_I, shown in Figure 2.1, are the interior domain, exterior domain and interface, respectively. The elliptic interface problem is given by

$$-\nabla \cdot \beta \nabla u = f \quad \text{in } \Omega_1 \cup \Omega_2, \tag{2.1}$$

$$u = g \quad \text{on } \partial\Omega, \tag{2.2}$$

$$u|_{\Omega_1} - u|_{\Omega_2} = a \quad \text{on } \Gamma_I, \tag{2.3}$$

$$((\beta\nabla u)|_{\Omega_1} - (\beta\nabla u)|_{\Omega_2}) \cdot \boldsymbol{n}_e = b \quad \text{on } \Gamma_I, \tag{2.4}$$

where f, g, a and b are functions of x and y, $\boldsymbol{n}_e$ is the outward unit normal vector to $\partial\Omega_1$ and β is a positive finite constant function on Ω_1 and Ω_2, separately. Here the source term f may be discontinuous across the interface.

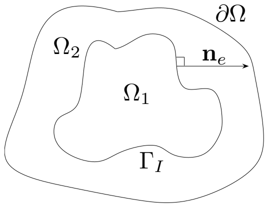

Figure 2.1: Domain and interface.

3. Numerical methods

3.1. Discontinuous Galerkin weak formulation

We first discretize the domain Ω and define $\mathcal{T}_h = \{K\}$ as a subdivision of Ω into non-overlapping quadrilaterals. Γ is the union of the boundaries of elements K in $\mathcal{T}_h$ with the assumption that the interface Γ_I is included in Γ; thus, the quadrilaterals are fitted to the interface. We use curvilinear quadrilateral elements for a curved boundary or interface. Γ_0 represents the set of interior edges in Γ that are not interface edges. Therefore, we have the decomposition of $\Gamma = \Gamma_0 \cup \Gamma_I \cup \partial\Omega$.

To solve an elliptic interface problem using the LDG method, we introduce an auxiliary variable $\boldsymbol{q}$ and write problem (2.1)–(2.4) into a first order system:

$$\begin{aligned} -\nabla\cdot\boldsymbol{q} &= f && \text{in } \Omega_1\cup\Omega_2,\\ \boldsymbol{q} &= \beta\nabla u && \text{in } \Omega_1\cup\Omega_2,\\ u &= g && \text{on } \partial\Omega,\\ u|_{\Omega_1} - u|_{\Omega_2} &= a && \text{on } \Gamma_I,\\ (\boldsymbol{q}|_{\Omega_1} - \boldsymbol{q}|_{\Omega_2})\cdot\boldsymbol{n}_e &= b && \text{on } \Gamma_I. \end{aligned}$$

Using the weak formulation of the first order system, we multiply equations $\boldsymbol{q} = \beta\nabla u$ and $-\nabla\cdot\boldsymbol{q} = f$ by test functions $\boldsymbol{v}$ and w, respectively, integrate the results over element K and use integration by parts to obtain

$$\int_K \frac{1}{\beta}\boldsymbol{q}\cdot\boldsymbol{v}\,dx + \int_K u\nabla\cdot\boldsymbol{v}\,dx - \int_{\partial K} u\boldsymbol{v}\cdot\boldsymbol{n}_K\,ds = 0, \tag{3.1}$$

$$\int_K \nabla w\cdot\boldsymbol{q}\,dx - \int_{\partial K} w\boldsymbol{q}\cdot\boldsymbol{n}_K\,ds = \int_K fw\,dx, \tag{3.2}$$

where $\boldsymbol{n}_K$ is the outward unit normal vector to the ∂K.

We derive the discontinuous Galerkin formulation by replacing the edge values of u in (3.1) and $\boldsymbol{q}$ in (3.2) with the modified LDG flux [14],

$$\widehat{\boldsymbol{q}}_{K,e}(u_h, \boldsymbol{q}_h) = \begin{cases} \{\!\{\boldsymbol{q}_h\}\!\} - \frac{\eta}{h_e}[\![u_h]\!] & \text{if } e \subseteq \Gamma_0, \\ \{\!\{\boldsymbol{q}_h\}\!\} - \frac{\eta}{h_e}[\![u_h]\!] + \frac{\eta}{h_e} a\boldsymbol{n}_e + \frac{1}{2}b\boldsymbol{n}_K & \text{if } e \subseteq \Gamma_I, \\ \boldsymbol{q}_h - \frac{\eta}{h_e}u_h\boldsymbol{n} + \frac{\eta}{h_e}g\boldsymbol{n} & \text{if } e \subseteq \partial\Omega, \end{cases} \tag{3.3}$$

$$\widehat{u}_{K,e}(u_h, \boldsymbol{q}_h) = \begin{cases} \{\!\{u_h\}\!\} & \text{if } e \subseteq \Gamma_0, \\ \{\!\{u_h\}\!\} + \frac{1}{2}a\boldsymbol{n}_e \cdot \boldsymbol{n}_K & \text{if } e \subseteq \Gamma_I, \\ g & \text{if } e \subseteq \partial\Omega, \end{cases} \tag{3.4}$$

where η is the penalization parameter and h_e the length of edge e. The notations $[\![\cdot]\!]$ and $\{\!\{\cdot\}\!\}$ are jump and average operators in the DG methods. Consider two adjacent elements K^+ and K^- in Ω and $e = \overline{\partial K^+} \cap \overline{\partial K^-}$ as the common edge. For a scalar function w and a vector function $\boldsymbol{v}$ defined on $\overline{K^+} \cup \overline{K^-}$, we define $[\![w]\!]$ and $[\![\boldsymbol{v}]\!]$ on $e = \overline{\partial K^+} \cap \overline{\partial K^-}$ as

$$\begin{aligned} [\![w]\!] &= w|_{K^+}\boldsymbol{n}^+ + w|_{K^-}\boldsymbol{n}^-, \\ [\![\boldsymbol{v}]\!] &= \boldsymbol{v}|_{K^+} \cdot \boldsymbol{n}^+ + \boldsymbol{v}|_{K^-} \cdot \boldsymbol{n}^-, \end{aligned}$$

where $\boldsymbol{n}^+$ and $\boldsymbol{n}^-$ are the outward units normal to ∂K^+ and ∂K^-, respectively. On the other hand, we define the average operator $\{\!\{\cdot\}\!\}$ as

$$\begin{aligned} \{\!\{w\}\!\} &= \frac{1}{2}(w|_{K^+} + w|_{K^-}), \\ \{\!\{\boldsymbol{v}\}\!\} &= \frac{1}{2}(\boldsymbol{v}|_{K^+} + \boldsymbol{v}|_{K^-}). \end{aligned}$$

The discontinuous Galerkin formulation takes the standard form: Find u in $H^1(\mathcal{T}_h)$ and $\boldsymbol{q}$ in $\left[H^1(\mathcal{T}_h)\right]^2$, such that for all K in $\mathcal{T}_h$,

$$\begin{aligned} \int_K \frac{1}{\beta}\boldsymbol{q} \cdot \boldsymbol{v}\, dx + \int_K u\nabla \cdot \boldsymbol{v}\, dx - \int_{\partial K} \widehat{u}_{K,e}\boldsymbol{v} \cdot \boldsymbol{n}_K\, ds = 0, &\qquad \forall\, \boldsymbol{v} \in \left[H^1(\mathcal{T}_h)\right]^2, \\ \int_K \nabla w \cdot \boldsymbol{q}\, dx - \int_{\partial K} w\widehat{\boldsymbol{q}}_{K,e} \cdot \boldsymbol{n}_K\, ds = \int_K fw\, dx, &\qquad \forall\, w \in H^1(\mathcal{T}_h). \end{aligned}$$

We then approximate $H^1(\mathcal{T}_h)$ and $\left[H^1(\mathcal{T}_h)\right]^2$ by the spaces $\mathcal{V}_h$ and $\mathcal{M}_h$ defined by

$$\begin{aligned} \mathcal{V}_h &= \left\{w \in L^2(\Omega) : w|_K \in Q^\ell(K) \text{ for all } K \text{ in } \mathcal{T}_h\right\}, \\ \mathcal{M}_h &= \left\{\boldsymbol{v} \in \left[L^2(\Omega)\right]^2 : \boldsymbol{v}|_K \in [Q^\ell(K)]^2 \text{ for all } K \text{ in } \mathcal{T}_h\right\}, \end{aligned}$$

where $Q^\ell(K)$ is the space of bi-polynomial functions of degree at most $\ell \geq 1$ on K. The discretization of the LDG weak problem is then given as follows:

Problem 3.1. Find u_h in $\mathcal{V}_h$ and $\boldsymbol{q}_h$ in $\mathcal{M}_h$ such that for all K in $\mathcal{T}_h$,

$$\int_K \frac{1}{\beta}\boldsymbol{q}_h \cdot \boldsymbol{v}\, dx + \int_K u_h \nabla \cdot \boldsymbol{v}\, dx - \int_{\partial K} \widehat{u}_{K,e}\boldsymbol{v}\cdot \boldsymbol{n}_K\, ds = 0, \qquad \forall\, \boldsymbol{v} \in [Q^\ell(K)]^2, \tag{3.5}$$

$$\int_K \nabla w \cdot \boldsymbol{q}_h\, dx - \int_{\partial K} w\widehat{\boldsymbol{q}}_{K,e}\cdot \boldsymbol{n}_K\, ds = \int_K fw\, dx, \quad \forall\, w \in Q^\ell(K), \tag{3.6}$$

where $\widehat{u}_{K,e}$ and $\widehat{\boldsymbol{q}}_{K,e}$ are defined in (3.3) and (3.4)

3.2. Mixed form of LDG methods

Because DG methods are in fact mixed finite element methods, the approximation solution $(\boldsymbol{q}_h, u_h)$ of LDG discretization (3.5) and (3.6) can be characterized as the solution of the variational problem as follows.

Find $(\boldsymbol{q}_h, u_h) \in \mathcal{M}_h \times \mathcal{V}_h$ such that

$$a_h(\boldsymbol{q}_h, \boldsymbol{v}) + b_h(u_h, \boldsymbol{v}) = G_h(\boldsymbol{v}), \tag{3.7}$$

$$-b_h(w, \boldsymbol{q}_h) + c_h(u_h, w) = F_h(w), \tag{3.8}$$

where

$$\begin{aligned}
a_h(\boldsymbol{q}_h, \boldsymbol{v}) &= \sum_{K\in\mathcal{T}_h}\int_K \frac{1}{\beta}\boldsymbol{q}_h\cdot\boldsymbol{v}\, dx,\\
b_h(u_h, \boldsymbol{v}) &= \sum_{K\in\mathcal{T}_h}\int_K u_h\nabla\cdot\boldsymbol{v}\, dx - \int_{\Gamma_0\cup\Gamma_I}\{\!\!\{u_h\}\!\!\}\,[\![\boldsymbol{v}]\!]\, ds\\
&= -\sum_{K\in\mathcal{T}_h}\int_K \nabla u_h\cdot\boldsymbol{v}\, dx + \int_{\Gamma_0\cup\Gamma_I}[\![u_h]\!]\cdot\{\!\!\{\boldsymbol{v}\}\!\!\}\, ds + \int_{\partial\Omega} u_h\boldsymbol{v}\cdot\boldsymbol{n}\, ds,\\
b_h(w, \boldsymbol{q}_h) &= -\sum_{K\in\mathcal{T}_h}\int_K \nabla w\cdot\boldsymbol{q}_h\, dx + \int_{\Gamma_0\cup\Gamma_I}[\![w]\!]\cdot\{\!\!\{\boldsymbol{q}_h\}\!\!\}\, ds + \int_{\partial\Omega} w\boldsymbol{q}_h\cdot\boldsymbol{n}\, ds,\\
c_h(u_h, w) &= \int_{\Gamma_0\cup\Gamma_I}\frac{\eta}{h_e}[\![w]\!]\cdot[\![u_h]\!]\, ds + \int_{\partial\Omega}\frac{\eta}{h_e}wu_h\, ds,\\
G_h(\boldsymbol{v}) &= \int_{\Gamma_I}\{\!\!\{\boldsymbol{v}\}\!\!\}\cdot a\boldsymbol{n}_e\, ds + \int_{\partial\Omega} g\boldsymbol{v}\cdot\boldsymbol{n}\, ds,\\
F_h(w) &= \int_\Omega fw\, dx + \int_{\Gamma_I}\frac{\eta}{h_e}[\![w]\!]\cdot a\boldsymbol{n}_e\, ds + \int_{\Gamma_I}\{\!\!\{w\}\!\!\}\, b\, ds + \int_{\partial\Omega}\frac{\eta}{h_e}gw\, ds.
\end{aligned}$$

As a result, the corresponding linear system has the form

$$\begin{bmatrix} A & B\\ -B^t & C\end{bmatrix}\begin{bmatrix} Q\\ U\end{bmatrix} = \begin{bmatrix} G\\ F\end{bmatrix}, \tag{3.9}$$

where A, B and C are matrices associated with the bilinear forms a_h, b_h and c_h, respectively. G and F are vectors associated with the functions G_h and F_h, respectively. Q and U are vectors containing the expansion coefficients of the numerical solutions $\boldsymbol{q}_h$ and u_h.

Note that A is block diagonal and symmetric, B is antisymmetric and C is symmetric.

3.3. Matrix reduction

In the previous subsection, the discretization of the LDG weak problem was written as a linear system (3.9).

When the basis functions are orthogonal, matrix A is diagonal, and we can easily rewrite the system into smaller systems. For a general quadrilateral element, the basis functions are not orthogonal, and we can derive the reduced linear system using the Uzawa method [2]. For some matrix operations,

$$\begin{bmatrix} I & O \\ B^tA^{-1} & I \end{bmatrix} \begin{bmatrix} A & B \\ -B^t & C \end{bmatrix} \begin{bmatrix} I & -A^{-1}B \\ O & I \end{bmatrix} \begin{bmatrix} I & A^{-1}B \\ O & I \end{bmatrix} \begin{bmatrix} Q \\ U \end{bmatrix} = \begin{bmatrix} I & O \\ B^tA^{-1} & I \end{bmatrix} \begin{bmatrix} G \\ F \end{bmatrix}$$

or

$$\begin{bmatrix} A & O \\ O & B^tA^{-1}B + C \end{bmatrix} \begin{bmatrix} Q + A^{-1}BU \\ U \end{bmatrix} = \begin{bmatrix} G \\ B^tA^{-1}G + F \end{bmatrix}.$$

The above mentioned system is equivalent to

$$\begin{cases} Q = A^{-1}G - A^{-1}BU, \\ (B^tA^{-1}B + C)U = B^tA^{-1}G + F. \end{cases}$$

Let $X = A^{-1}G$ and $Y = A^{-1}B$. We then rewrite the system as

$$\begin{cases} A[X\ Y] = [G\ B], \\ Q = X - YU, \\ (B^tY + C)U = B^tX + F. \end{cases} \tag{3.10}$$

Set $X = [X_1, X_2, \ldots, X_p]^t$, $Y = [Y_1, Y_2, \ldots, Y_p]^t$, $G = [G_1, G_2, \ldots, G_p]^t$ and $B = [B_1, B_2, \ldots, B_p]^t$. Because A is block diagonal and symmetric, $A = \text{diag}(A_1, A_2, \ldots, A_p)$, where each A_i is symmetric for $i = 1, 2, \ldots, p$. We then solve the reduced linear system (3.10) as follows:

Step 1. Solve for X_i and Y_i in the subsystem

$$A_i[X_i\ Y_i] = [G_i\ B_i] \tag{3.11}$$

for $i = 1, 2, \ldots, p$. From this, we can obtain X and Y.

Step 2. Solve for U in the subsystem

$$(B^tY + C)U = B^tX + F. \tag{3.12}$$

Step 3. Using X, Y and U, we compute $Q = X - YU$.

Because the size of each matrix A_i is less than $B^tY + C$, the overall computation time is primarily dominated by the computation time required to solve for U in subsystem (3.12).

Because the linear system (3.9) has three times as many degrees of freedom as the linear system (3.12), solving the reduced linear system is more efficient when compared with the origin system, as illustrated in Section 4.1.2.

3.4. Bilinear blended transfinite map and quadrature grid points

To solve two-dimensional problems using the LDG method, we compute numerical integrations over the quadrilateral elements using the Legendre-Gauss-Lobatto (LGL) quadrature grid points x_i, y_j and weights w_i, w_j. Over the closed interval $[-1,1]$, the LGL grid points x_k are defined as the zeros of the polynomials $(1-x^2)P'_n(x)$, where $P'_n(x)$ is the derivative of the n-th degree Legendre polynomial and the weights w_k of the grid points x_k are defined as

$$w_k = \frac{2}{n(n-1)[P_{n-1}(x_k)]^2}.$$

On the square domain $S = [-1,1] \times [-1,1]$, we denote the LGL grid points as follows:

$$\begin{aligned} \{(x_i, y_j) \mid -1 = x_0 < x_1 < \cdots < x_i < \cdots < x_N = 1, \\ -1 = y_0 < y_1 < \cdots < y_j < \cdots < y_M = 1\}, \end{aligned}$$

as shown in Figure 3.2. Suppose that a general curvilinear quadrilateral domain Q is a closed, bounded, and simply connected region in the xy-plane with its boundary ∂Q divided into four parametric curve segments. To construct the integral nodes in Q, we use the bilinearly blended transfinite map [13] to rearrange the LGL grid points in S.

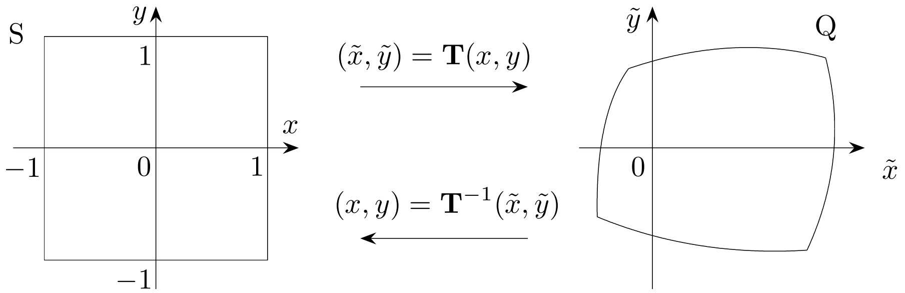
S
y
1
x
-1
0
1
-1
$(\tilde{x}, \tilde{y}) = \mathbf{T}(x, y)$
$(x, y) = \mathbf{T}^{-1}(\tilde{x}, \tilde{y})$
$\tilde{y}$
Q
0
$\tilde{x}$

Figure 3.1: Bilinear blended transfinite map transforms the square region onto a general quadrilateral element.

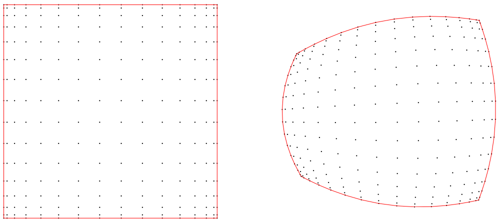

Figure 3.2: Bilinear blended transfinite map transforms the integral nodes in the square region S (left) onto a general quadrilateral element Q (right).

We now describe how to construct the map from S to Q, as shown in Figure 3.1. Suppose that the boundary ∂Q is constituted by the four parametric curves $\boldsymbol{F}(-1,y)$, $\boldsymbol{F}(1,y)$, $\boldsymbol{F}(x,-1)$ and $\boldsymbol{F}(x,1)$ from ∂S to ∂Q. Define $\boldsymbol{T}(x,y)$ to be a one-to-one mapping from the square domain S in the xy-plane onto the region Q in the $\widetilde{x}\widetilde{y}$-plane, such that

$$\begin{aligned}\boldsymbol{T}(x,y) &= \begin{bmatrix}\widetilde{x}(x,y)\\ \widetilde{y}(x,y)\end{bmatrix}\\ &= \frac{1}{2}(1-y)\boldsymbol{F}(x,-1) + \frac{1}{2}(1+y)\boldsymbol{F}(x,1)\\ &\quad + \frac{1}{2}(1-x)\boldsymbol{F}(-1,y) + \frac{1}{2}(1+x)\boldsymbol{F}(1,y)\\ &\quad - \frac{1}{4}(1+y)(1-x)\boldsymbol{F}(-1,1) - \frac{1}{4}(1+y)(1+x)\boldsymbol{F}(1,1)\\ &\quad - \frac{1}{4}(1-y)(1-x)\boldsymbol{F}(-1,-1) - \frac{1}{4}(1-y)(1+x)\boldsymbol{F}(1,-1).\end{aligned} \tag{3.13}$$

We then construct the grid points $\widetilde{x}_i$ and $\widetilde{y}_j$ in Q, as shown in Figure 3.2. Using the integral nodes and weights in S and Q, we compute the numerical integration on Q as follows:

$$\begin{aligned}\iint_Q f(\widetilde{x},\widetilde{y})\,d\widetilde{x}d\widetilde{y} &= \iint_S [f\circ T](x,y)\,|J(x,y)|\,dxdy,\\ &= \int_{-1}^{1}\int_{-1}^{1} f(\widetilde{x}(x,y),\widetilde{y}(x,y))\,|J(x,y)|\,dxdy\\ &\approx \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} f(\widetilde{x}(x_i,y_j),\widetilde{y}(x_i,y_j))\,|J(x_i,y_j)|\,w_iw_j,\end{aligned} \tag{3.14}$$

where $J(x,y)$ is the Jacobian matrix of mapping (3.13).

3.5. Quadrature rule for curvilinear quadrilateral elements

For curvilinear quadrilateral elements, mapping (3.13) will introduce nonlinearity in (3.14). To resolve the issue, recall Theorem 2.10 in [6]. Using P^k-polynomial basis functions and triangular elements, to achieve $(k+1)$st-order accuracy, the quadrature rule over the edges is exact for polynomials of degree $(2k+1)$ and the quadrature rule over the elements is exact for polynomials of degree $2k$.

Although we cannot apply the theorem in our implementation directly, the theorem gives us some idea to choose the quadrature rule. Using Q^k-polynomials basis functions, we expect $(k+1)$st-order accuracy and $k+2$ LGL integration points, including end points, are needed on each edge.

4. Numerical experiments

In this section, we undertake some numerical experiments to study the performance of our scheme on an elliptic interface problem:

$$\begin{aligned} -\nabla\cdot\beta\nabla u &= f \quad \text{in } \Omega_1\cup\Omega_2,\\ u &= g \quad \text{on } \partial\Omega,\\ u|_{\Omega_1} - u|_{\Omega_2} &= a \quad \text{on } \Gamma_I,\\ ((\beta\nabla u)|_{\Omega_1} - (\beta\nabla u)|_{\Omega_2})\cdot\boldsymbol{n}_e &= b \quad \text{on } \Gamma_I. \end{aligned}$$

In the following experiments, we describe only the domain, the interface, the media property and the exact solution of each problem. From the media property and the exact solution, we can easily derive the corresponding jump conditions a and b, Dirichlet condition g, and source term f. The penalization parameter η in (3.3) and (3.4) is set to 1.

Before continuing, we briefly introduce the numerical experiments. In test Problem 1, we use the curved-edge elements to solve an elliptic interface problem with a circular interface. In test Problem 2, we use the curved-edge elements to solve an elliptic interface problem with a complicated interface.

To study performance, we compute the errors of the numerical solution U_{num} in L^2 norm numerically as follows:

$$L^2\text{-error of } u = \sqrt{\sum_{K\in\mathcal{T}_h}\left(\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\left[U_{\text{num}}(x_i,y_j)-U_{\text{exa}}(x_i,y_j)\right]^2|J(x_i,y_j)|\,w_iw_j\right)}$$

where x_i, y_j, and w_i, w_j are the nodes and weights, respectively, of the LGL quadrature, as described in Section 3.4. Using Q^k-polynomial basis functions, we expect $(k+1)$st-order accuracy and $k+2$ integration points, including end points, are needed on each edge. In addition, the convergence order is estimated using the least-square method.

We implemented our numerical experiments using the MATLAB programming language. All numerical tests were performed on an ordinary desktop equipped with an Intel(R) Core(TM) i7 930 2.80 GHz CPU.

4.1. Test Problem 1

In this test, we study the performance of curved elements for the curved interface problem Figure 4.1. We assume that $\Omega = [-2,2] \times [-2,2]$ is the whole domain, the open unit disk Ω_1 with center at the origin is the interior domain, $\Omega_2 = \Omega \setminus \overline{\Omega}_1$ is the exterior domain, and $\Gamma_I(\theta) = (\cos(\theta), \sin(\theta))$ for $\theta \in [0, 2\pi)$ is the interface. The media property is

$$\begin{cases} 1 & \text{in } \Omega_1, \\ 10 & \text{in } \Omega_2. \end{cases}$$

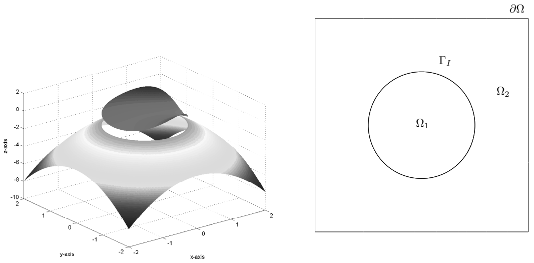

Figure 4.1: The numerical solution and the computational domain of test problem 1.

The exact solution is

$$u(x,y) = \begin{cases} e^x(y^2 + x^2 \sin y) & \text{in } \Omega_1, \\ -(x^2+y^2) & \text{in } \Omega_2. \end{cases}$$

The corresponding jump conditions a and b, Dirichlet condition g and the source term f can be derived easily.

Two typical curved-edge meshes are presented in Figure 4.2.

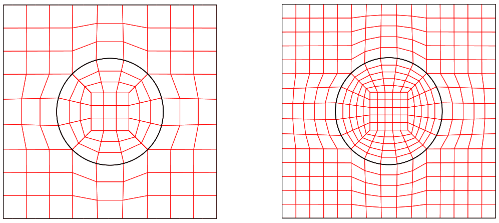

Figure 4.2: Curved-edge quadrilateral meshes with 117 (left) and 325 (right) elements of test problem 1.

4.1.1. Convergence

The h-convergence results shown in Table 4.1 and Figure 4.3 indicate that u converges with order $k+1$ in L^2-norm with Q^k-polynomials. The results also show that $\boldsymbol{q}$ converges with order k except Q^1-polynomials. The results also show that our choice of quadrature rules, described in Section 3.4, are appropriate for curvilinear quadrilateral elements.

In addition, we studied the p-convergence behavior on a specific mesh. The computational domain is shown in Figure 4.4 (right). Table 4.2 and Figure 4.4 (left) show that the numerical solution converges exponentially.

L^2-errors of u

	Number of elements						
Deg	13	52	117	208	325	468	order
1	$8.34e-01$	$3.40e-01$	$1.84e-01$	$1.12e-01$	$7.30e-02$	$5.07e-02$	1.94
2	$1.30e-01$	$1.01e-02$	$3.59e-03$	$1.37e-03$	$6.61e-04$	$3.64e-04$	3.23
3	$2.44e-02$	$2.46e-03$	$3.56e-04$	$1.14e-04$	$4.44e-05$	$2.04e-05$	4.01
4	$1.29e-02$	$2.48e-04$	$4.61e-05$	$8.36e-06$	$2.46e-06$	$9.00e-07$	5.28
5	$2.22e-03$	$5.01e-05$	$2.57e-06$	$5.93e-07$	$1.47e-07$	$4.69e-08$	6.05
6	$9.90e-04$	$5.83e-06$	$4.67e-07$	$4.02e-08$	$7.60e-09$	$1.89e-09$	7.32

L^2-errors of $\boldsymbol{q}$

	Number of elements						
Deg	13	52	117	208	325	468	order
1	$1.06e+01$	$9.10e+00$	$7.01e+00$	$5.86e+00$	$5.09e+00$	$4.51e+00$	0.64
2	$2.50e+00$	$2.41e-01$	$1.03e-01$	$3.67e-02$	$1.76e-02$	$9.93e-03$	3.04
3	$6.08e-01$	$8.11e-02$	$1.27e-02$	$4.69e-03$	$2.21e-03$	$1.22e-03$	3.55
4	$5.43e-01$	$9.94e-03$	$2.59e-03$	$4.58e-04$	$1.33e-04$	$4.86e-05$	5.10
5	$7.20e-02$	$2.73e-03$	$1.40e-04$	$3.40e-05$	$9.97e-06$	$3.79e-06$	5.61
6	$6.52e-02$	$3.52e-04$	$3.87e-05$	$3.40e-06$	$6.57e-07$	$1.66e-07$	7.12

Table 4.1: h-convergence of u and $\boldsymbol{q}$ for test problem 1.

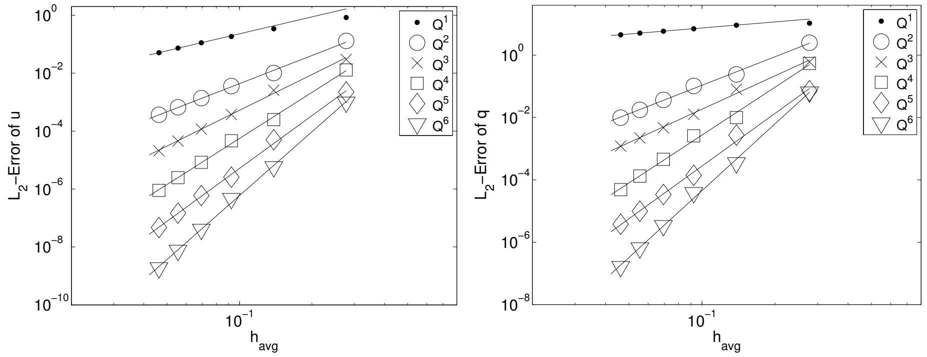

Figure 4.3: h-convergence of u (left) and $\boldsymbol{q}$ (right) for test problem 1.

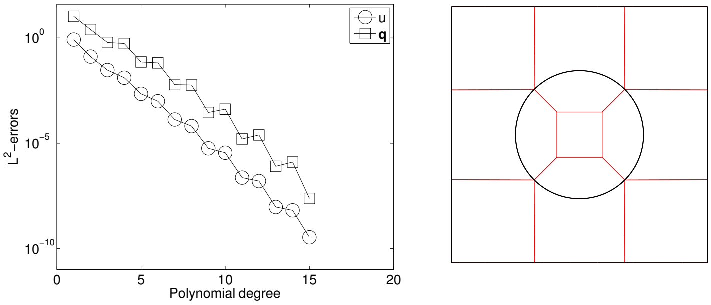

Figure 4.4: p-convergence of u and $\boldsymbol{q}$ on a 13-element mesh (left). Corresponding curved-edge quadrilateral mesh with 13 elements (right) in test problem 1.

Basis function	Degrees of freedom	L^2-error of u	L^2-error of $\boldsymbol{q}$
Q^1	156	$8.337e-01$	$1.059e+01$
Q^2	351	$1.298e-01$	$2.503e+00$
Q^3	624	$2.967e-02$	$6.076e-01$
Q^4	975	$1.289e-02$	$5.425e-01$
Q^5	1404	$2.221e-03$	$7.202e-02$
Q^6	1911	$9.899e-04$	$6.518e-02$
Q^7	2496	$1.343e-04$	$6.068e-03$
Q^8	3159	$6.514e-05$	$5.790e-03$
Q^9	3900	$5.728e-06$	$2.884e-04$
Q^{10}	4719	$3.534e-06$	$4.149e-04$
Q^{11}	5616	$2.335e-07$	$1.595e-05$
Q^{12}	6591	$1.609e-07$	$2.477e-05$
Q^{13}	7644	$9.445e-09$	$8.036e-07$
Q^{14}	8775	$6.546e-09$	$1.283e-06$
Q^{15}	9984	$3.433e-10$	$2.417e-08$

Table 4.2: p-convergence of L^2-errors of u and $\boldsymbol{q}$ (13 elements) for test problem 1.

4.1.2. Efficiency and Performance of Matrix Reduction

In Table 4.3, we list the computation times and corresponding L^2-errors of u by degrees of freedom, where

- TG is the computation time required to generate all matrices of the linear system (3.9).
- TL is the computation time required to solve the original linear system (3.9).
- TS is the computation time required to solve the reduced linear system (3.10).

It is observed that the Uzawa method described in Section 3.3 improves the computational performance significantly. The computation time required to solve the reduced linear system (3.10) is about a third of the time required to solve the original one (3.9), i.e., TL / TS $\approx$ 3.

Table 4.3 also suggests that given finite computational resources (which are related to the matrix size or degrees of freedom), the best strategy for solving the problem is to use the highest possible polynomial degree on the coarsest mesh.

DoF	No. of elements	Basis function	L^2-errors of u	TG(s)	TL(s)	TS(s)
1404	13	Q^5	$+2.220e-03$	1.95	0.10	0.038
	52	Q^2	$+1.011e-02$	0.90	0.106	0.038
	117	Q^1	$+1.841e-01$	0.80	0.095	0.045
5616	13	Q^{11}	$+2.3348e-07$	26.44	3.32	1.12
	52	Q^5	$+5.007e-05$	7.25	3.33	1.04
	117	Q^3	$+3.709e-04$	3.80	3.36	1.04
	208	Q^2	$+1.372e-03$	3.02	3.34	1.05
	468	Q^1	$+5.071e-02$	2.50	3.33	1.09
9984	13	Q^{15}	$+3.4328e-10$	85.62	17.23	5.35
	52	Q^7	$+7.552e-07$	22.84	18.30	5.25
	208	Q^3	$+1.171e-04$	7.20	17.21	5.03
	832	Q^1	$+2.792e-02$	4.52	17.12	5.14

Table 4.3: Comparison of accuracies for given degrees of freedom for test problem 1. TG, TL and TS represent the time required to generate all matrices of the linear system, to solve the linear system and to solve the reduced linear system, respectively.

4.2. Test Problem 2

In this test, we consider a problem, taken from [14, 17], with a curved interface. In Figure 4.5, we assume that $\Omega = [-1,1] \times [0,3]$ is the entire domain and Ω_1 is the open interior domain embedded in Ω with a complicated interface

$$\Gamma_I(\theta) = \begin{pmatrix} 0.6\cos\theta - 0.3\cos 3\theta \\ 1.5 + 0.7\sin\theta - 0.07\sin 3\theta + 0.2\sin 7\theta \end{pmatrix}$$

for $\theta \in [0, 2\pi]$. The exterior domain is $\Omega_2 = \Omega \setminus \Omega_1$.

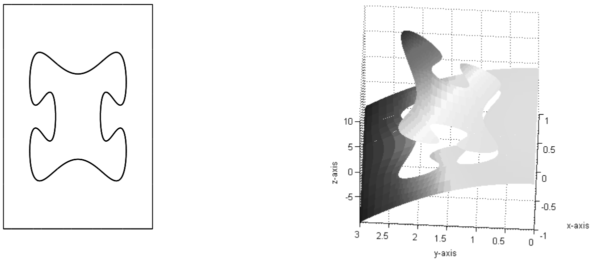

Figure 4.5: Domain (left) and numerical solution (right) of test problem 2.

The media property is given by $\beta = \begin{cases} 1 & \text{in } \Omega_1, \\ 10 & \text{in } \Omega_2. \end{cases}$ The discontinuous exact solution is

$$u(x,y) = \begin{cases} e^x(y^2 + x^2\sin y) & \text{in } \Omega_1, \\ -(x^2+y^2) & \text{in } \Omega_2. \end{cases}$$

The corresponding jump conditions a and b, Dirichlet condition g, and source term f can be derived easily.

To solve the problem, we divided the domain into a mesh with curved elements and studied the p-convergence history, as shown in Table 4.4 and Figure 4.6. To obtain a numerical solution with L_2-error less than 0.1, we required only 25 elements with Q^3 polynomial space (degrees of freedom = 1200). In contrast, Guyomarc'h et al. [14] required 60, 069 straight-edge triangular elements (with P^2 polynomial space).

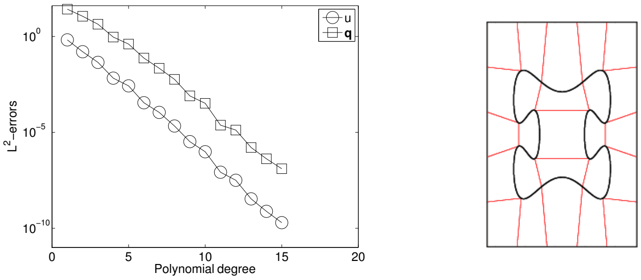

Figure 4.6: p-convergence of u and $\boldsymbol{q}$ on a 25-element mesh (left). Corresponding curved-edge quadrilateral mesh with 25 elements (right).

Basis function	DoF	L^2-error of u	L^2-error of $\boldsymbol{q}$
Q^1	300	$6.615e-01$	$2.641e+01$
Q^2	675	$1.599e-01$	$1.129e+01$
Q^3	1200	$4.450e-02$	$4.346e+00$
Q^4	1875	$6.711e-03$	$9.273e-01$
Q^5	2700	$2.652e-03$	$4.010e-01$
Q^6	3675	$3.512e-04$	$7.475e-02$
Q^7	4800	$1.095e-04$	$2.204e-02$
Q^8	6075	$2.148e-05$	$5.815e-03$
Q^9	7500	$3.297e-06$	$7.967e-04$
Q^{10}	9075	$9.766e-07$	$3.258e-04$
Q^{11}	10800	$8.334e-08$	$2.376e-05$
Q^{12}	12675	$3.172e-08$	$1.316e-05$
Q^{13}	14700	$3.406e-09$	$1.621e-06$
Q^{14}	16875	$7.608e-10$	$4.190e-07$
Q^{15}	19200	$1.955e-10$	$1.269e-07$

Table 4.4: p-convergence of u and $\boldsymbol{q}$ (25 elements) for test problem 2.

5. Conclusions and remarks

In this study, we implemented the high-order LDG method to solve the elliptic interface problem.

For accuracy, we used curvilinear quadrilateral elements with Q^k-polynomial basis functions to solve elliptic interface problems with curved interfaces and boundaries. We selected LGL quadrature rules with $k+2$ integration points, including end points, on each edge. The results of numerical experiments confirm that quadrature rules werw appropriate for curvilinear quadrilateral elements.

The results also show that the numerical solution converges with the order $k+1$. Furthermore, we obtained the exponential p-convergence in our numerical results.

We also implemented the Uzawa method to rewrite the original linear system into smaller linear systems. The resulting method is three times faster than the original one.

Moreover, because high-order methods are more efficient than low-order methods, we can conclude that under the condition of limited computational resources, the best approach for achieving optimal performance is to solve a problem using local spaces with the highest-degree polynomials and the coarsest mesh, which also reduces the cost of mesh generation.

Acknowledgments

This work was supported in part by the National Science Council of Taiwan via the grant NSC 98-2115-M-006-007. The authors would like to thank Professor Ming-Chih Lai for suggesting this problem. The authors also thank Professor Chern-Shuh Wang for the helpful discussion.

References

[1] D. N. Arnold, F. Brezzi, B. Cockburn and L. D. Marini, *Unified analysis of discontinuous Galerkin methods for elliptic problems*, SIAM J. Numer. Anal. **39** (2002), no. 5, 1749–1779. http://dx.doi.org/10.1137/s0036142901384162

[2] K. Arrow, L. Hurwicz and H. Uzawa, *Studies in Nonlinear Programming*, Stanford University Press, Stanford, CA, 1958.

[3] I.-L. Chern and Y.-C. Shu, *A coupling interface method for elliptic interface problems*, J. Comput. Phys. **225** (2007), no. 2, 2138–2174. http://dx.doi.org/10.1016/j.jcp.2007.03.012

[4] B. Cockburn, *Discontinuous Galerkin methods*, ZAMM Z. Angew. Math. Mech. **83** (2003), no. 11, 731–754. http://dx.doi.org/10.1002/zamm.200310088

[5] B. Cockburn, J. Gopalakrishnan and R. Lazarov, *Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems*, SIAM J. Numer. Anal. **47** (2009), no. 2, 1319–1365. http://dx.doi.org/10.1137/070706616

[6] B. Cockburn, S. Hou and C.-W. Shu, *The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case*, Math. Comp. **54** (1990), no. 190, 545–581. http://dx.doi.org/10.2307/2008501

[7] B. Cockburn, S. Y. Lin and C.-W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems*, J. Comput. Phys. **84** (1989), no. 1, 90–113. http://dx.doi.org/10.1016/0021-9991(89)90183-6

[8] B. Cockburn and C.-W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework*, Math. Comp. **52** (1989), no. 186, 411–435.
http://dx.doi.org/10.1090/s0025-5718-1989-0983311-4

[9] ______, *The Runge-Kutta local projection P^1-discontinuous-Galerkin finite element method for scalar conservation laws*, RAIRO Modél. Math. Anal. Numér. **25** (1991), no. 3, 337–361. http://dx.doi.org/10.2514/6.1988-3797

[10] ______, *The Runge-Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems*, J. Comput. Phys. **141** (1998), no. 2, 199–224.
http://dx.doi.org/10.1006/jcph.1998.5892

[11] ______, *The local discontinuous Galerkin method for time-dependent convection-diffusion systems*, SIAM J. Numer. Anal. **35** (1998), no. 6, 2440–2463.
http://dx.doi.org/10.1137/s0036142997316712

[12] ______, *Runge-Kutta discontinuous Galerkin methods for convection-dominated problems*, J. Sci. Comput. **16** (2001), no. 3, 173–261.
http://dx.doi.org/10.1023/A:1012873910884

[13] W. J. Gordon and C. A. Hall, *Transfinite element methods: Blending-function interpolation over arbitrary curved element domains*, Numer. Math. **21** (1973), no. 2, 109–129. http://dx.doi.org/10.1007/bf01436298

[14] G. Guyomarc'h, C.-O. Lee and K. Jeon, *A discontinuous Galerkin method for elliptic interface problems with application to electroporation*, Comm. Numer. Methods Engrg **25** (2009), no. 10, 991–1008. http://dx.doi.org/10.1002/cnm.1132

[15] J. L. Hellrung, L. Wang, E. Sifakis and J. M. Teran, *A second order virtual node method for elliptic problems with interfaces and irregular domains in three dimensions*, J. Comput. Phys. **231** (2012), no. 4, 2015–2048.
http://dx.doi.org/10.1016/j.jcp.2011.11.023

[16] R. J. LeVeque and Z. L. Li, *The immersed interface method for elliptic equations with discontinuous coefficients and singular sources*, SIAM J. Numer. Anal. **31** (1994), no. 4, 1019–1044. http://dx.doi.org/10.1137/0731054

[17] X.-D. Liu, R. P. Fedkiw and M. Kang, *A boundary condition capturing method for Poisson's equation on irregular domains*, J. Comput. Phys. **160** (2000), no. 1, 151–178. http://dx.doi.org/10.1006/jcph.2000.6444

[18] X.-D. Liu and T. C. Sideris, *Convergence of the ghost fluid method for elliptic equations with interface*, Math. Comp. **72** (2003), no. 244, 1731–1746.
http://dx.doi.org/10.1090/s0025-5718-03-01525-4

Min-Hung Chen
Department of Mathematics, National Cheng Kung University, Tainan 701, Taiwan
E-mail address: minhung@gmail.com

Rong-Jhao Wu
Institute of Aeronautics and Astronautics, National Cheng Kung University, Tainan 701, Taiwan
E-mail address: wrj0618@gmail.com