Construction of Jacobian Matrix for Solving Convection-Diffusion Problem with Interior Penalty Method

Liang Li and Songping Wu*
School of Aeronautic Science and Engineering, Beihang University, Beijing, 100191, People’s Republic of China
E-mail: wusping825@163.com

Abstract. In this paper we describe in details constructing the Jacobian matrix based on the interior penalty discontinuous Galerkin discretization of the time-dependent convection-diffusion problem, which is requisite for the nonlinear solver when an implicit method is used in time integration. Special attention is paid to the treatment of periodic boundary conditions during the construction. The convergence rate of the nonlinear solver is tested in numerical simulation to illustrate the utility of the Jacobian matrix.

1. Introduction

Semi-discretization is widely used in discontinuous Galerkin (DG) method. For which, the global solution of the problem is approximated by the direct sum of local solutions. Spatial and temporal variables are separated in the local solution. After multiplying the global approximation by test functions and integrating the products by parts, a system of ordinary differential equations (ODEs) for the degrees of freedom of the approximation is formed. In [1], explicit total variation diminishing (TVD) Runge-Kutta method is used to solve the ODEs, laying the foundation of a popular family of DG methods known as Runge-Kutta discontinuous Galerkin (RKDG).

For guarantying stability, RKDG methods suffer from a severe limitation on the step length in time integration [1, 2]. Alternatively, implicit schemes are unconditionally stable for solving ODEs. Therefore, it is natural to combine DG spatial discretization and implicit temporal discretization to overcome the restriction of time step length. In implicit schemes, nonlinear equations need to be solved, and the Newton-Raphson method is frequently chosen as the solver. It enjoys a super-linear convergence rate on condition that the Jacobian matrix for nonlinear equations is accurate enough and the initial guess is sufficiently close to the solution.

Interior penalty (IP) method is a kind of DG methods applicable to solving diffusion problems. It can be dated back to the 1970s [3-5]. A brief introduction of its development can be found in chapter 1 of [6]. Unlike its two competitors, i.e., Bassi-Rebay (BR) method and local DG (LDG) method [7, 8], no auxiliary quantity approximating the gradient of the primal variable is needed in IP method, delivering it a significant advantage regarding calculating efficiency.

In this paper, for solving the time-dependent convection-diffusion problem with a combination of non-symmetric IP method and diagonally implicit Runge-Kutta method, we provide a detailed description on the construction of global Jacobian matrix. Particular attention is paid to the treatment of periodic boundary conditions during the construction.
2. Model problem and spatial discretization

We are concerned with the following time-dependent convection-diffusion problem

\[
\begin{align*}
\frac{\partial u}{\partial t} + \nabla \cdot ( \mathbf{F} u ) &= \varepsilon \Delta u, & (x,t) \in \Omega \times (0,T] \\
\mathbf{F} &= \mathbf{F}(x), & x \in \partial\Omega_D \\
\mathbf{n} \cdot \nabla u &= g_N, & x \in \partial\Omega_N \\
u(x,0) &= u_0(x), & x \in \Omega
\end{align*}
\]  

(1)

where \( \Omega \) is a domain in \( \mathbb{R}^d \), \( d=1 \) or \( 2 \), \( \partial\Omega = \partial\Omega_D \cup \partial\Omega_N \). \( u_0(x) \) is given, representing Dirichlet boundary conditions, Neumann boundary conditions and initial conditions, respectively.

Let us introduce some notation first. We call \( h_T \) a triangulation of \( \Omega \) when the domain is partitioned into \( K \) non-overlapping elements. \( \partial\Omega_k \) is referred to the boundary of element \( \Omega_k \). We further define the unions of element faces as

\[
\begin{align*}
F_I &= \{ \Gamma : \Gamma = \partial\Omega^k \cap \partial\Omega^l, \ \partial\Omega^k \cap \partial\Omega^l \subset \mathbb{R}^{d-1}, \ \Omega^k, \Omega^l \in T_h \}, \\
F_D &= \{ \Gamma : \Gamma = \partial\Omega^k \cap \partial\Omega_D, \ \partial\Omega^k \cap \partial\Omega_D \subset \mathbb{R}^{d-1}, \ \Omega^k \in T_h \}, \\
F_N &= \{ \Gamma : \Gamma = \partial\Omega^k \cap \partial\Omega_N, \ \partial\Omega^k \cap \partial\Omega_N \subset \mathbb{R}^{d-1}, \ \Omega^k \in T_h \}.
\end{align*}
\]

Through the paper we assume \( T_h \) is conforming, i.e., for \( \Omega^k \in T_h \), the result of \( \Omega^k \) is either \( \emptyset \), or a vertex, or an element of \( F_I \). In the last case the two elements are called neighbors.

We pursue the approximate solution of the modal problem in the space of broken piecewise polynomial functions

\[
S_{hp} = \{ v : v \in L^2(\Omega), \ v|_{\partial\Omega^k} \in P_p(\Omega^k), \ \Omega^k \in T_h \},
\]

where \( P_p(\Omega^k) \) refers to the set of polynomials of degree \( \leq p \) with \( \Omega^k \) as support. We choose Lagrange interpolation functions \( \ell_i^k(x), i=1,2,\cdots,N_p \) defined on evenly distributed nodes on \( \Omega^k \) as the basis functions for \( P_p(\Omega^k) \) [9]. It is obvious the union of \( \ell_j^k(x), j=1,2,\cdots,N_p \), \( \Omega^k \in T_h \) constitutes a basis for \( S_{hp} \). By referring \( v^k_{in} \) and \( v^k_{ex} \) to the interior and exterior trace of \( v \in S_{hp} \) on \( \Omega^k \), respectively, we define the mean and the jump over the boundary of \( \Omega^k \) as

\[
\langle v \rangle = \frac{1}{2}(v^k_{in} + v^k_{ex}), \quad [[v]] = v^k_{in} - v^k_{ex}.
\]

The prescription of the value of \( v^k_{ex} \) on \( \partial\Omega \) will be discussed later.

We approximate \( u \) on \( \Omega^k \) with

\[
u^k_h = \sum_{j=1}^{N_p} u_{j}^k(t) \ell_{j}^k(x),
\]

(2)

where unknowns \( u_{j}^k(t) \) can be viewed as an approximation of \( u_{j}(x,t) \). Moreover, we suppose the global approximate solution of equation (1) to be the direct sum of the \( K \) local expressions, i.e.,

\[
u_h = \bigoplus_{k=1}^{K} u_h^k.
\]

(3)

Substituting equation (3) into equation (1), multiplying the resultant with \( \ell_{j}(x) \), applying the standard techniques of IP method in [6] and rewriting the consequent from the perspective of element, we obtain the identity
\[ \int_{\Omega} \frac{du^k_i}{dt} l^k_i \, d\Omega = C^k_i + D^k_i . \]  

Note that different from the conventional face-centered expression of IP method, equation (4) is element-centered. It can better reflect the (relatively) independence of the local approximation, an essential character of the DG method. It further benefits vectorization and parallel computing concerning the practical application.

In equation (4), the convection-related term is defined as

\[ C^k_i = \int_{\Omega} \mathbf{F}(u^k_i) \cdot \nabla l^k_i \, d\Omega - \int_{\partial\Omega} H(u^k_i, u^k_{ex}, \mathbf{n}^k) l^k_i \, dS . \]  

\[ H(u^k_{in}, u^k_{ex}, \mathbf{n}^k) \] is known as numerical flux and a widely used type of which is local Lax-Friedrichs flux

\[ H(u^k_{in}, u^k_{ex}, \mathbf{n}^k) = \frac{1}{2} \mathbf{n}^k \cdot (\mathbf{F}(u^k_{in}) + \mathbf{F}(u^k_{ex})) + \frac{\lambda}{2} (u^k_{in} - u^k_{ex}) , \quad \lambda = \max_{[\min(u^k_{in}, u^k_{ex})], \max(u^k_{in}, u^k_{ex})}] \left| \mathbf{n}^k \cdot \frac{d\mathbf{F}}{du} \right| . \]  

where \( \mathbf{n}^k \) denotes the unit outward normal vector to \( \partial\Omega^k \) and \( u^k_{ex} \) on \( \partial\Omega \) is prescribed using extrapolation.

The diffusion-related term \( D^k_i \) differs as the location of element \( \Omega^k \). If \( \partial\Omega^k \subset F_1 \cup F_2 \), it is defined as

\[ D^k_i = -\int_{\Omega} \varepsilon \nabla u^k_i \cdot \nabla l^k_i \, d\Omega + \int_{\partial\Omega^k} \mathbf{n}^k \cdot (\varepsilon \nabla u^k_i) l^k_i - \sigma([\varepsilon u^k_i]) l^k_i + \Theta([\varepsilon u^k_i]) \mathbf{n}^k \cdot \frac{1}{2} \nabla l^k_i \, dS , \]  

where the exterior value on \( F_D \) is prescribed using \( u_D \), i.e.,

\[ \langle \varepsilon \nabla u^k_i \rangle_{\partial\Omega^k} = \frac{1}{2} \left( \varepsilon \nabla u^k_{in} + \varepsilon \nabla u_D \right) , \quad \langle [\varepsilon u^k_i] \rangle_{\partial\Omega^k} = \varepsilon u^k_{in} - \varepsilon u_D . \]  

On the other hand, assuming a face of \( \Omega^k \), e.g., \( \partial\Omega^k \cap 2 \), coincides with \( \partial\Omega_N \), then

\[ D^k_i = -\int_{\Omega} \varepsilon \nabla u^k_i \cdot \nabla l^k_i \, d\Omega + \int_{\partial\Omega^k} \mathbf{n}^k \cdot (\varepsilon \nabla u^k_i) l^k_i - \sigma([\varepsilon u^k_i]) l^k_i + \Theta([\varepsilon u^k_i]) \mathbf{n}^k \cdot \frac{1}{2} \nabla l^k_i \, dS + \int_{\partial\Omega^k} \varepsilon \mathbf{g}^k l^k_i \, dS . \]  

In equations (7) and (8), parameter \( \Theta \) decides to which variant the DG scheme belongs, i.e., symmetric IP (SIPG) method (\( \Theta=1 \)), incomplete IP (IIPG) method (\( \Theta=0 \)), or non-symmetric IP (NIPG) method (\( \Theta=-1 \)). Penalty weight \( \sigma \) is defined as

\[ \sigma^k = C_u (h_i)^{-1} , \quad \Gamma \in F_1 \cup F_D , \]  

where \( h_i \) represents a \((d-1)\)-dimensional measurement of \( \Gamma \). Constant \( C_u \) has to be chosen neither too small to sustain the stability of the solution nor too big to maintain the efficiency of the calculation for SIPG and IIPG [10]. However, for NIPG, there is no difficulty choosing \( C_u \) since any positive \( \sigma \) is competent to keep the stability of the approximation. We take \( \Theta = -1 \) and \( C_u = 1 \) in this paper.

Letting \( i \) take each value in \( \{1,2,\cdots,N_p\} \) for every \( k \in \{1,2,\cdots,K\} \) in equation (4), we thus get a system of ODEs

\[ \frac{d\mathbf{U}}{dt} = \mathbf{M} \mathbf{U} + \mathbf{R} \equiv \mathbf{F} \]  

where

\[ \mathbf{M} = \begin{pmatrix} 
M^1 & 0 & \cdots & 0 \\
0 & M^2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & M^K 
\end{pmatrix}, \quad \mathbf{U} = \begin{pmatrix} 
\mathbf{U}^1 \\
\mathbf{U}^2 \\
\vdots \\
\mathbf{U}^K 
\end{pmatrix}, \quad \text{and} \quad \mathbf{R} = \begin{pmatrix} 
\mathbf{R}^1 \\
\mathbf{R}^2 \\
\vdots \\
\mathbf{R}^K 
\end{pmatrix}. \]
For $\forall k \in \{1, 2, \ldots, K\},$

$$M^k = \begin{bmatrix}
\int_{\Omega} I_{kk} t_{kk} d\Omega & \cdots & \int_{\Omega} I_{kk} t_{kk} d\Omega \\
\vdots & \ddots & \vdots \\
\int_{\Omega} I_{kk} N_{kk} d\Omega & \cdots & \int_{\Omega} I_{kk} N_{kk} d\Omega
\end{bmatrix}, \quad U^k = \begin{bmatrix}
u^k \\
\vdots \\
u^k_N
\end{bmatrix},$$

and $R^i$ corresponds to $C^i + D^i$ with subscript $i$ ranging from 1 to $N_p$ in equation (4).

### 3. Time integration and nonlinear solver

We integrate equation (9) using a three-stage, third-order accurate, L-stable diagonally implicit Runge-Kutta scheme in the form:

\[
\begin{align*}
U_i &= U_n + \Delta t \sum_{j=0}^{3} a_j \mathbf{F}(t_u + c_j \Delta t, U_j) \\
U_{n+1} &= U_n + \Delta t \sum_{j=0}^{3} b_j \mathbf{F}(t_u + c_j \Delta t, U_j)
\end{align*}
\]

where $\Delta t$ represents the length of the integral step and the coefficients are given in the following Butcher tableau [11]:

\[
\begin{array}{c|ccc}
\alpha & \alpha & 0 & 0 \\
\gamma & -\alpha & \alpha & 0 \\
\beta_1 & \beta_2 & \alpha & \\
\end{array}
\quad
\begin{cases}
\alpha = 0.435866521508459 \\
\beta_1 = -(6\alpha^2 - 16\alpha + 1) / 4 \\
\beta_2 = (6\alpha^2 - 20\alpha + 5) / 4 \\
\gamma = (\alpha + 1) / 2
\end{cases}
\]

In every stage of equation (10), a set of nonlinear equations needs to be solved. Taking $i=1$ as an example, we have

\[
G(U_1) \equiv U_n + \alpha \Delta t \mathbf{F}(t_u + \alpha \Delta t, U_i) - U_i = 0.
\]

An inexact Newton-Raphson method

\[
\begin{bmatrix}
\frac{dG}{dU}
\end{bmatrix}_{U_{1,m}} \Delta U_{1,m} = -\omega_m G\left(U_{1,m}\right), \quad 0 < \omega_m \leq 1
\]

is used to solve the problem [12]. One can find that the expressions of Jacobian matrices are same for all stages in equation (10), i.e.,

\[
\frac{dG}{dU} = \alpha \Delta t M^{-1} \frac{dR}{dU} - I,
\]

with $I$ referring to identity matrix.

### 4. Construction of Jacobian matrix

From equation (12), it is clear that the Jacobian matrix can be easily achieved once $J \equiv \frac{dR}{dU}$ is known.

In this section we will describe its construction in details.

There are $(N_p \times K)^2$ quantities in $J$. The matrix can be divided into $K^2$ squares of size $(N_p)^2$ and the $(m, n)$-th square, denoted by $J^{m,n}$, corresponds to $\frac{dR^m}{dU^n}$, see equation (9) for reference. Considering equations (4) - (8), the right-hand side of equation (4) involves information of $\Omega^k$ and its
neighbors only, suggesting in most cases $J^{k,n}$ is empty, with the exceptions $n = k$ or element $\Omega^k$ is a neighbor of $\Omega^k$.

We first consider the expression of $J^{k,k}$. The $(i, j)$-th value of the square is defined by taking the partial derivative of $C^k_i + D^k_i$ with respect to $u^k_j$ in equation (4), i.e.,

$$J^{k,k}_{ij} = \frac{\partial C^k_i}{\partial u^k_j} + \frac{\partial D^k_i}{\partial u^k_j}. \quad (13)$$

The first partial derivative is defined as

$$\frac{\partial C^k_i}{\partial u^k_j} = \int_{\Omega^k} \frac{dF}{du^k_j} \cdot \nabla l^k_i \, d\Omega - \frac{\partial}{\partial u^k_j} \left( \int_{\Omega^k} H l^k_i \, dS \right). \quad (14)$$

Special attention shall be paid to the last term in equation (14). In equation (6), both the maximum and the absolute value functions are non-differentiable. Nonetheless, we can numerically approximate the non-differentiable part of $H$ using difference quotient, i.e., by defining

$$H_{non} \left( u^k_{in}, u^k_{ex} \right) = \frac{\lambda}{2} \left( u^k_{in} - u^k_{ex} \right),$$

we have

$$\left[ \frac{\partial}{\partial u^k_j} \left( \int_{\Omega^k} H_{non} l^k_i \, dS \right) \right] = \frac{1}{\delta} \int_{\Omega^k} \left[ H_{non} \left( \sum_{j=1}^N u^k_j \delta l^k_j + (u^k_j + \delta) l^k_j, u^k_{ex} \right) - H_{non} \left( u^k_{in}, u^k_{ex} \right) \right] l^k_i \, dS \quad (15)$$

$$\left[ \frac{\partial}{\partial u^k_j} \left( \int_{\Omega^k} H l^k_i \, dS \right) \right] = \int_{\Omega^k} \frac{1}{2} \frac{dF}{\partial u^k_j} \big|_{u^k_j} l^k_i \, dS + \frac{\partial}{\partial u^k_j} \left( \int_{\Omega^k} H_{non} l^k_i \, dS \right)$$

where $\delta$ is a small positive number.

By introducing a switch function

$$\Phi(x) = \begin{cases} 1, & x \in \Gamma \cup F_D \\ 0, & x \in \Gamma \in F_N \end{cases} \quad (16)$$

the partial derivative of the diffusion-related term in equation (13) can be expressed as

$$\frac{\partial D^k_i}{\partial u^k_j} = -\int_{\Omega^k} \epsilon \nabla l^k_i \cdot \nabla l^k_i \, d\Omega + \int_{\Omega^k} \Phi(x) \left( \frac{1}{2} \epsilon \nabla l^k_i \cdot \nabla l^k_i - \sigma \epsilon l^k_i l^k_i - \Theta \epsilon l^k_i l^k_i - \frac{1}{2} \nabla l^k_i \right) \, dS. \quad (17)$$

Next we concentrate on the blocks away from the principle diagonal of $J$. We use superscript $knbr$ to denote the $r$-th neighbor of element $\Omega^k$. Taking $knbr$ as an example, we get

$$J^{k,knbr}_{ij} = \frac{\partial C^k_i}{\partial u^k_{knbr}} + \frac{\partial D^k_i}{\partial u^k_{knbr}}. \quad (18)$$

where

$$\frac{\partial C^k_i}{\partial u^k_{knbr}} = -\int_{\Omega^k} H l^k_i \, dS \quad (19)$$

and

$$\frac{\partial D^k_i}{\partial u^k_{knbr}} = \int_{\Omega^k} \Phi(x) \left( \frac{1}{2} \epsilon \nabla l^k_{knbr} \cdot \nabla l^k_i + \sigma \epsilon l^k_{knbr} l^k_i - \Theta \epsilon l^k_{knbr} l^k_i - \frac{1}{2} \nabla l^k_i \right) \, dS. \quad (20)$$

Similar to equation (15), equation (19) is approximated with difference quotient, i.e.,
Equations (13) - (21) are competent to construct Jacobian matrix for implicitly solving problems with Dirichlet and Neumann boundary conditions. When it comes to periodic boundary conditions, extra effort shall be taken. It is necessary to embody periodic boundary conditions in the construction of \( \mathbf{J} \) since the approximate solution exchanges information with itself on boundary subjected to the conditions. Contrarily, Dirichlet or Neumann boundary conditions have nothing to do with the Jacobian matrix in that neither \( \mathbf{u}^D \) nor \( \mathbf{u}^N \) is introduced in \( \mathbf{J} \), see equations (14), (15), (17), (19) and (20) for reference.

We illustrate our method with a one-dimensional example,

\[
\begin{align*}
\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = \varepsilon \frac{\partial^2 u}{\partial x^2}, & \quad (x,t) \in [0,L] \times (0,T) \\
\mathbf{u}(0,t) = \mathbf{u}(L,t), \quad \mathbf{u}(L,t) = \mathbf{u}(0,t), \quad \mathbf{u}(x,t) = \mathbf{u}_0(x)
\end{align*}
\]

According to Section 2, we divide \( \Omega = [0,L] \) into \( K \) elements and approximate \( u \) with equation (2). To bring in the impact of periodic boundary conditions, we create fictitious elements on the ends of \( \Omega \). That is realized by copying the element on \( \partial \Omega \) and the associated local approximation and translating them over a distance of \( L \). A sketch of the procedure can be found in figure 1, where the fictitious cells are marked by \( K+1 \) and \( K+2 \).

![Figure 1. Creation of fictitious elements.](image)

Thanks to fictitious elements, the influence of periodic boundary conditions is introduced into \( \mathbf{J} \) because \( \mathbf{J}^{1,K} \) and \( \mathbf{J}^{K,1} \) are not empty. Specifically,

\[
\mathbf{J}^{1,K}_y = \frac{\partial C^i}{\partial u^K_j} + \frac{\partial D^i}{\partial u^{K+1}_j} = \frac{\partial C^i}{\partial u^{K+1}_j} = \mathbf{J}^{1,K+1}_y,
\]

and

\[
\mathbf{J}^{K,1}_y = \frac{\partial C^K_j}{\partial u^1_i} + \frac{\partial D^K_j}{\partial u^{K+2}_i} = \frac{\partial C^K_j}{\partial u^{K+2}_i} = \mathbf{J}^{K,K+2}_y.
\]

The calculation of \( \mathbf{J}^{1,K+1} \) and \( \mathbf{J}^{K,K+2} \) obeys equations (18) - (21) while the value of function \( \Phi(x) \) in equation (16) is one on boundaries subjected to periodic boundary conditions.

5. Numerical examples

Several numerical examples are given in this section and each of them is computed on a sequence of refined meshes. To test the performance of the Jacobian matrix, during the computation on the finest
mesh, we record iterative residuals in the first stage for the beginning time steps [13]. The termination criteria for the Newton-Raphson method is 1e-8 in our applications. According to equation (11), linear equations need to be solved in every iteration for the Newton-Raphson method. We use a generalized minimum residual (GMRES) solver with incomplete LU factorizations as preconditioners to deal with the problem. In numerically solving the convection-diffusion problem, we define the step length for time integration as

$$\Delta t^k = \min \left\{ \rho^k \left( \frac{dF}{du} \right)_k^{-1}, \ (\rho^k)^2 \varepsilon^{-1} \right\}, \quad \Delta t = \frac{C}{2p+1} \varepsilon \min \left\{ \Delta t^k \right\}$$  (22)

where $\rho^k$ denotes the diameter of the inscribed ball for element $\Omega^k$ and $C$ is an adjustable parameter of order one.

5.1. One-dimensional Burgers equation [15]

$$\begin{cases}
    u_t + 0.5u^2 = \varepsilon u_{xx}, & x \in (0,2) \\
    u(x,0) = 2\varepsilon \pi \sin(\pi x) \left[ \sigma + \cos(\pi x) \right]^{-1}, & \varepsilon = 0.1, \quad \sigma = 2 \\
    u(0,t) = u(2,t), \quad u(2,t) = u(0,t)
\end{cases}$$

The analytic solution of the problem is

$$u(x,t) = 2\varepsilon \pi \exp\left(-\pi^2 \varepsilon t \sin(\pi x) \left[ \sigma + \exp(-\pi^2 \varepsilon t \cos(\pi x) \right]^{-1}.$$

In numerically solving the problem, uniform meshes with 32, 64, 128 and 256 elements are successively taken as triangulation. We choose $p=3$ for local approximations and run each simulation up to $t=0.1$. In order to decide a suitable time step length, we first solve the problem with a combination of the IP method and explicit, three-stage, third-order accurate, TVD Runge-Kutta method. We observe that the scheme is unstable unless parameter $C$ in equation (22) is set smaller than 0.1. Next, we use equation (10) as temporal discretization with $5C=5$ in the definition of time step length.

Table 1 shows the $L^2$ errors and their experimental order of convergence (EOC) for the numerical solutions achieved by the implicit scheme. It is evident that the theoretically optimal $(p+1)$-th order accuracy of the DG method is well achieved. Table 2 presents the nonlinear residuals during iterations. We find that after only one iteration the order of the residual decreases from 1e-5 to 1e-9, demonstrating the super-linear convergence rate of the Newton-Raphson method, and thus the accuracy of the Jacobian matrix.

| $K$ | Error | EOC |
|-----|-------|-----|
| 32  | 3.403e-6 | -   |
| 64  | 1.869e-7 | 4.19|
| 128 | 1.114e-8 | 4.07|
| 256 | 7.143e-10| 3.96|

| Iteration | Step 1 Res | Factor | Step 2 Res | Factor | Step 3 Res | Factor | Step 4 Res | Factor |
|-----------|-----------|-------|-----------|-------|-----------|-------|-----------|-------|
| 0         | 9.987e-5  | -     | 9.977e-5  | -     | 9.968e-5  | -     | 9.959e-5  | -     |
| 1         | 7.440e-9  | 13423 | 7.440e-9  | 13410 | 7.439e-9  | 13400 | 7.438e-9  | 13389 |
5.2. Two-dimensional linear convection-diffusion equation

\[
\begin{align*}
    u_i + v(u_x + u_y) &= \varepsilon(u_{xx} + u_{yy}), \quad (x,y) \in (0,1) \times (0,1) \\
    u(x,y,0) &= \sin(2\pi(x+y)), \quad v = 1, \quad \varepsilon = 0.05 \\
    u(0,y,t) &= u(1,y,t), \quad u(x,0,t) = u(x,1,t)
\end{align*}
\]

The analytic solution of the problem is

\[
u(x,y,t) = \exp(-8\pi^2\varepsilon t)\sin(2\pi(x+y) - 4\pi\varepsilon t).
\]

In figure 2 we show a series of unstructured meshes with uniform triangular elements, where fictitious elements used for dealing with periodic boundary conditions are presented in dashed lines for the coarsest one. The mesh with 64 × 64 × 2 elements is not shown because the grids in which are too dense to recognize. Similar to example 5.1, \( p = 3 \), \( t = 0.1 \), and explicit scheme is first used as the temporal discretization to set a reference for time step length. We find that the solution is unstable with \( C \) bigger than 0.15. Next, we take \( C = 1 \) in defining step length for implicit time integration.

Table 3 presents the \( L^2 \) errors of the numerical solutions achieved by implicit scheme, indicating the optimal \((p+1)\)-th order accuracy of the DG method. Table 4 shows the iterative residuals during computations on the finest mesh. We find that in every time step the order of the residual decreases from 1e-3 to 1e-11 after two iterations. An explanation of the high convergence rate is that the governing equation in example 5.2 is linear with respect to \( u \), thus the Jacobian matrix is independent with \( U \) and remains unchanged during iterations. Considering equation (11), when \( U_n \) is taken as the initial guess, one iteration is able to get \( U_1 \) if the matrix is precise. It is the linear solver deciding the magnitude of the residual. The high speed of convergence implies the accuracy of the Jacobian matrix.

![Refined meshes of example 5.2.](a) \( K = 8 \times 8 \times 2 \) (b) \( K = 16 \times 16 \times 2 \) (c) \( K = 32 \times 32 \times 2 \)

**Table 3.** \( L^2 \) errors of example 5.2.

| \( K \) | Error | EOC |
|--------|-------|-----|
| 8 \times 8 \times 2 | 3.482e-4 | - |
| 16 \times 16 \times 2 | 2.543e-5 | 3.78 |
| 32 \times 32 \times 2 | 1.681e-6 | 3.92 |
| 64 \times 64 \times 2 | 1.086e-7 | 3.95 |

**Table 4.** Residuals and their reduction factors of example 5.2, \( K = 64 \times 64 \times 2 \).

| Iteration | Step 1 | Step 2 | Step 3 | Step 4 |
|-----------|--------|--------|--------|--------|
|           | Res    | Factor | Res    | Factor | Res    | Factor | Res    | Factor |
| 0         | 0.0014 | -      | 0.0014 | -      | 0.0014 | -      | 0.0014 | -      |
5.3. Two-dimensional inviscid Burgers equation

\[
\begin{align*}
&
\frac{\partial u}{\partial t} + \left( \frac{1}{2} \left( u^2 \right)_x + \frac{1}{2} \left( u^2 \right)_y \right) = 0, \quad (x, y) \in (-1,1) \times (-1,1) \\
&
u(x, y, 0) = 0.25 + 0.5\sin(\pi(x + y)) \\
&u(-1, y, t) = u(1, y, t), \quad u(x, -1, t) = u(x, 1, t)
\end{align*}
\]

The exact solution of the problem obeys the identity:

\[
u(x, y, t) = 0.25 + 0.5\sin(\pi(x + y - 2ut)).
\]

Since there is no diffusion terms in the governing equation, the Jacobian matrix can be easily achieved by setting \( \varepsilon = 0 \) for all diffusion-related equations in section 2. Similar to example 5.2, unstructured meshes with uniform triangular elements are taken as triangulations of the domain. From the coarsest mesh to the finest one, \( K \) equals \( 16 \times 16 \times 2 \), \( 32 \times 32 \times 2 \) and \( 64 \times 64 \times 2 \), respectively. We choose \( p = 2 \) for local approximations and run each simulation up to \( t = 0.2 \), when the shock has not appeared in the solution. In this example, time step length is defined by

\[
\Delta t^k = \min \left\{ \rho^k \left( \left\| \frac{dF}{du} \right\| \right)^{-1} \right\}, \quad \Delta t = \frac{C}{2p + 1} \min \{ \Delta t^k \}.
\]

For RKDG method, stability requires \( C \leq 1 \) in above equation, see [1] and [2] for reference. We set \( C = 5 \) for the implicit temporal discretization in our simulation.

Table 5 presents the \( L^2 \) errors between the numerical and exact solutions on different meshes. Compared to the results in table 1 and table 3, there is a minor detriment to the optimal \((p+1)\)-th order accuracy for DG method. Similar to example 5.1, although the governing equation is nonlinear with respect to \( u \), thus the numerical flux is non-differentiable, the results in table 6 have reflected the super-linear convergence rate for the Newton-Raphson method, implying the precision of the Jacobian matrix.

| \( K \)     | Error | EOC |
|------------|-------|-----|
| \( 16 \times 16 \times 2 \) | 0.0062 | -   |
| \( 32 \times 32 \times 2 \) | 9.327e-4 | 2.74 |
| \( 64 \times 64 \times 2 \) | 1.393e-4 | 2.74 |

Table 6. Residuals and their reduction factors of example 5.3, \( K = 128 \times 128 \times 2 \).

| Iteration | Step 1 Res | Factor | Step 2 Res | Factor | Step 3 Res | Factor | Step 4 Res | Factor |
|-----------|------------|--------|------------|--------|------------|--------|------------|--------|
| 0         | 0.0104     | -      | 0.0109     | -      | 0.0114     | -      | 0.0120     | -      |
| 1         | 7.020e-6   | 1482   | 7.589e-6   | 1436   | 8.203e-6   | 1390   | 9.090e-6   | 1320   |
| 2         | 5.075e-8   | 138    | 5.028e-8   | 151    | 6.007e-8   | 137    | 6.704e-8   | 136    |
| 3         | 4.734e-10  | 107    | 5.044e-10  | 99     | 5.971e-10  | 101    | 6.331e-10  | 106    |

6. Conclusions

Jacobian matrix is the core for implicitly integrating the IP discretization of time-dependent convection-diffusion problem. In this paper, we describe in details of its construction. Most work of the construction is based on analytically taking partial derivatives while the non-differentiable part is numerically approximated with difference quotient. A technique for dealing with periodic boundary conditions during the construction is proposed. In numerical test, the realization of the super-linear convergence rate of the Newton-Raphson method in solving nonlinear equations demonstrates the
accuracy of the Jacobian matrix, thus the utility of our construction.

Acknowledgements
This research was financially supported by the National Natural Science Foundation of China No. 91530325.

References
[1] Cockburn, B., Shu, C. W. (1989) TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework. Mathematics of Computation, 52: 411-435.
[2] Kubatko, E. J., Dawson, C., Westerink, J. J. (2008) Time step restrictions for Runge-Kutta discontinuous Galerkin methods on triangular grids. J. Comput. Phys., 227: 9697-9710.
[3] Douglas, J., Dupont, T. (1976) Interior penalty procedures for elliptic and parabolic Galerkin methods. Lecture Notes in Phys., 58: 207-216.
[4] Wheeler, M. F. (1978) An elliptic collocation-finite element method with interior penalties. SIAM J. Numer. Anal., 15: 152-161.
[5] Arnold, D. N. (1982) An interior penalty finite element method with discontinuous elements. SIAM J. Numer. Anal., 19: 742-760.
[6] Dolejší, V., Feistauer, M. (2015) Discontinuous Galerkin Method: Analysis and Applications to Compressible Flow. Springer Publishing Company, Incorporated.
[7] Bassi, F., Rebay, S. (1997) A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. J. Comput. Phys., 131: 267-279.
[8] Cockburn, B., Shu, C. W. (1998) The local discontinuous Galerkin method for time-dependent convection-diffusion systems. SIAM J. Numer. Anal., 35: 199–224.
[9] Hesthaven, J. S., Warburton, T. (2008) Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications. Springer Publishing Company, Incorporated.
[10] Shabahzi, K. (2005) An explicit expression for the penalty parameter of the interior penalty method. J. Comput. Phys., 205: 401-407.
[11] Persson, P. O. (2013) Shock capturing for high-order discontinuous Galerkin simulation of transient flow problems. In: 21th AIAA Computational Fluid Dynamics Conference. San Diego. pp. 1-9.
[12] Li, Q., Mo, Z., Qi, L. (1987) Numerical Methods for Nonlinear equations. China Science Publishing & Media Ltd, Beijing. (in Chinese)
[13] Hartmann, R. (2005) The role of the Jacobian in the adaptive discontinuous Galerkin method for the compressible Euler equations. In: Warnecke, G. (Eds.), Analysis and Numerics for Conservation Laws. Springer Publishing Company, Incorporated. pp. 301–316.
[14] Yang, S., Tao, W. (2006) Heat Transfer. Higher Education Press, Beijing. (in Chinese)
[15] Shao, L., Feng, X., He, Y. (2011) The discontinuous Galerkin finite element methods for Burger’s Equation. Mathematical & Computer Modelling., 54: 2943-2954.