A Reduced Order Local Discontinuous Galerkin Method for the Variable Coefficients Diffusion Equations

Danchen Zhu, Lingzhi Qian*, Jing Wang*

Abstract—**In this paper, the model reduction technique is applied to solve the variable coefficients diffusion equation. In the full order model (FOM), we use Crank-Nicolson (CN) method and local discontinuous Galerkin (LDG) method to discretize the variable coefficients diffusion equation. To construct the reduced order model (ROM), we use the proper orthogonal decomposition (POD) method and the Galerkin projection to reduce the number of unknowns in the FOM. Similar to FOM, the LDG method still can reach the optimal convergence in the ROM. At the same time, the ROM can provide accurate approximate solution of the variable coefficients diffusion equation with much less computational cost. Finally, some numerical tests are illustrated to confirm the performance of the proposed reduction method.**

Index Terms—**Local discontinuous Galerkin, Crank-Nicolson method, Variable coefficients diffusion equation, Proper orthogonal decomposition.**

I. Introduction

In this paper, we aim to construct a ROM of the variable coefficients diffusion equation, which can reduce the computational cost while maintaining high accuracy. The initial boundary value problem of variable coefficient diffusion is considered as follows

$$\begin{cases} \partial_t y(t,\boldsymbol{x}) + \nabla\cdot(b(t,\boldsymbol{x})\nabla y(t,\boldsymbol{x})) = f(t,\boldsymbol{x}) \text{ in } T\times\Omega, \\ y = y_D \text{ on } T\times\partial\Omega_{\mathrm{D}}(t), \\ -\nabla y\cdot\boldsymbol{n} = g_N \text{ on } T\times\partial\Omega_{\mathrm{N}}(t), \\ y = y^0 \text{ on } \{0\}\times\Omega, \end{cases} \tag{1}$$

where $T = (0, t_{end})$ is a finite time interval and $\Omega \in \mathbb{R}^2$ is a polygonally bounded domain with boundary $\partial\Omega$ subdivided into Dirichlet $\partial\Omega_D$ and Neumann $\partial\Omega_N$ parts. The function y_D, g_N and y^0 are Dirichlet boundary, Neumann boundary and initial data, respectively. The $\boldsymbol{n}(\boldsymbol{x})$ is the outward unit normal. Equation (1) is often applied to research the diffusive transport in fluids, the unknown $y(t,\boldsymbol{x})$ denotes the concentration of a solute, $b(t,\boldsymbol{x})$ is the diffusion coefficient and $f(t,\boldsymbol{x})$ accounts for generation or degradation of $y(t,\boldsymbol{x})$.

In order to obtain a set of high fidelity solutions of equation (1), we introduce the CN method and LDG method to obtain a FOM. In [13], Reed and Hill first time use the DG method to solve the neutron transport equation in 1973. Since then, the DG method has been made popular by Shu and Colkburn to solve the first-order hyperbolic conservative equation in [4]. The LDG method was proposed for the first time in [5] and further developed in [1]. The LDG method transforms the second-order equation into two first-order equations by introducing an auxiliary flux variable. The LDG method can handle general mesh [8] and is an ideal candidate for designing so-called h_p-adaptive solutions strategies [14]. However, under the same accuracy, the LDG method has far more degrees of freedom than other conforming finite elements method [11]. Therefore we introduce the model reduction technique to reduce the computational cost.

The main idea of model reduction is to construct a low-dimensional system that can provide accurate approximate solutions for the original high dimensional system. This method is required to use a low dimensional basis, which can be optimally constructed by an approach called proper orthogonal decomposition (POD). The POD method with the snapshot technique was introduced by Sirovich [15]. It is probably the most popular method for solving the POD basis [2]. This method with the Galerkin projection has been extensively used in numerous fields, such as analytical network modeling [16], electromagnetism [9], and flow dynamics [3], [6]. In particular, the ROM of DG method based on the POD method have been widely used for the partial differential equations to reduce the computational cost. For example, in order to solve the time-domain Maxwell equation, Li et al. proposed the ROM of DG method based on the POD method in [10]. The LDG method and POD method were first proposed by Polymarklam et al. to construct a reduced order model for solving the Burgers-poisson equation in [12]. In this paper, we introduce the POD method and CN-LDG method to construct the ROM for solving the variable coefficients diffusion equation.

This paper is organized as follows. In section 2, we propose the CN-LDG scheme to develop the FOM for solving the variable coefficients diffusion equation. In section 3, the ROM of the CN-LDG scheme based on the POD method is established. The numerical results in Section 4 demonstrate the effectiveness of the model reduction method for solving two-dimensional diffusion equations with variable coefficients. Some conclusions and future research are drawn in Section 5.

II. Full order model

In order to obtain a set of high fidelity solutions that can be used to construct POD basis, we first discretize the variable

Manuscript received April 17, 2023; revised September 07, 2023. This work was supported in part by the Natural Science Foundation of Guangxi (No. 2020GXNSFAA297223), Innovation Project of Guangxi Graduate Education, China (No.YCSW2023132) and the NSF of China (Nos.U19A2079, 11861054, 11671345 and 11771348).

Danchen Zhu is a graduate student of the Guangxi Normal University, Guilin Guangxi 541000, P.R. China (e-mail: zhudanchen@stu.gxnu.edu.cn).

* Corresponding author: Lingzhi Qian is an professor of the College of Mathematics and Statistics in Guangxi Normal University, Guilin 541006, P.R. China (corresponding author, e-mail: qianlz@mailbox.gxnu.edu.cn).

* Corresponding author: Jing Wang is a graduate student of the Guangxi Normal University, Guilin Guangxi 541000, P.R. China (corresponding author, e-mail: jingwang@stu.gxnu.edu.cn).

coefficients diffusion equation using CN-LDG method to obtain the FOM.

A. Spatial semi-discrete scheme

Let $\mathcal{S}_h = \{S\}$ be a regular family of non-overlapping partitions of Ω into K closed triangles S of characteristic size h such that $\overline{\Omega} = \cup S$. For $S \in \mathcal{S}_h$, let $\boldsymbol{n}_S$ denote the unit normal on ∂S exterior to S. Let ε_Ω denote the set of interior edges, and $\varepsilon_{\partial\Omega}$ denote the set of boundary edges, it can be further subdivided into Dirichlet ε_D and Neumann ε_N edges. And $\varepsilon = \varepsilon_\Omega \cup \varepsilon_{\partial\Omega}$ is the set of all edges.

For an interior edge $\xi \in \varepsilon_\Omega$ shared by triangles S^- and S^+, and for $\boldsymbol{x} \in \xi$, we define the one-side values of a scalar quantity $v = v(\boldsymbol{x})$ by

$$v^-(\boldsymbol{x}) = \lim_{\epsilon\to 0} v(\boldsymbol{x} - \epsilon \boldsymbol{n}_{S^-}) \text{ and } v^+(\boldsymbol{x}) = \lim_{\epsilon\to 0} v(\boldsymbol{x} - \epsilon \boldsymbol{n}_{S^+}), \quad (2)$$

but for a boundary edge $\xi \in \partial\Omega$, only the definition of the left-side value is meaningful. The average and jump of $v(\boldsymbol{x})$ on ξ are then given by

$$\{|v|\} = \frac{v^- + v^+}{2} \text{ and } [[v]] = v^-\boldsymbol{n}_{S^-} + v^+\boldsymbol{n}_{S^+} = (v^- - v^+)\boldsymbol{n}_{S^-}, \quad (3)$$

respectively. To formulate a LDG scheme, we first introduce an auxiliary variable $\boldsymbol{z} = -\nabla y(t, \boldsymbol{x})$ and equation (1) can be rewritten as follows

$$\begin{cases} \boldsymbol{z} = -\nabla y \text{ in } T \times \Omega, \\ \partial_t y + \nabla \cdot (b\boldsymbol{z}) = f \text{ on } T \times \Omega, \\ y = y_D \text{ on } T \times \Omega_D, \\ \boldsymbol{z} \cdot \boldsymbol{n} = g_N \text{ on } T \times \Omega_N, \\ y = y^0 \text{ on } \{0\} \times \Omega. \end{cases} \quad (4)$$

Let $\mathbb{P}_p(S)$ denote the space of polynomials of degree at most p on $S \in \mathcal{S}_h$. Therefore, the $\mathbb{P}_p(\mathcal{S}_h) = \Big\{v_h : \overline{\Omega} \to \mathbb{R}, \forall S \in \mathcal{S}_h, v_h|_S \in \mathbb{P}_p(S)\Big\}$ is the broken polynomial space on the triangulation $\mathcal{S}_h$. Incorporating the boundary conditions and adding penalty terms for the jumps in the primary unknowns, the semi-discrete scheme is given as follows:

For $t \in T, \forall S^- \in \mathcal{S}_h$ and $\forall \boldsymbol{w}_h \in [\mathbb{P}_p(\mathcal{S}_h)]^2$, $\forall v_h \in \mathbb{P}_P(\mathcal{S}_h)$, we can find $(\boldsymbol{z}_h(t), y_h(t)) \in [\mathbb{P}_p(\mathcal{S}_h)]^2 \times \mathbb{P}_p(\mathcal{S}_h)$ which holds

$$\begin{aligned} &\int_{S^-} \boldsymbol{w}_h \cdot \boldsymbol{z}_h(t) - \int_{S^-} \nabla \cdot \boldsymbol{w}_h y_h(t) + \int_{\partial S^-} \boldsymbol{w}_h^- \cdot \boldsymbol{n}_{S^-} \widehat{y}_h = 0 \\ &\int_{S^-} v_h \partial_t y_h(t) - \int_{S^-} \nabla v_h \cdot (b_h \boldsymbol{z}_h(t)) + \int_{\partial S^-} v_h^- \widehat{\boldsymbol{z}}_h = \int_{S^-} v_h f_h, \end{aligned} \quad (5)$$

where $\widehat{y}_h$ and $\widehat{\boldsymbol{z}}_h$ are give as follows

$$\widehat{y}_h = \begin{cases} \{|y_h(t)|\} \text{ on } \varepsilon_\Omega, \\ y_D(t) \text{ on } \varepsilon_{\Omega_D}, \\ y_h^-(t) \text{ on } \varepsilon_{\Omega_N}, \end{cases} \quad (6)$$

and

$$\widehat{\boldsymbol{z}}_h = \begin{cases} \{|b_h(t)\boldsymbol{z}_h(t)|\} \cdot \boldsymbol{n}_{S^-} + \dfrac{\eta}{h_{S^-}}[[y_h(t)]] \cdot \boldsymbol{n}_{S^-} \text{ on } \varepsilon_\Omega, \\ b_h^-(t)\boldsymbol{z}_h^-(t) \cdot \boldsymbol{n}_{S^-} + \dfrac{\eta}{h_{S^-}}(y_h^-(t) + y_D(t)) \text{ on } \varepsilon_{\Omega_D}, \\ b_h^-(t) g_N(t) \text{ on } \varepsilon_{\Omega_N}, \end{cases} \quad (7)$$

respectively, where $\eta > 0$ is the penalty coefficient, and h_{S^-} denotes the size of element S^-. Then, we denote a standard basis function $\varphi_{ki} : \overline{\Omega} \to \mathbb{R}$, which is only supported on the triangle $S_k \in \mathcal{S}_h$. The finite element space $\mathbb{P}_p(S_k)$ is denoted by

$$\mathbb{P}_P(S_k) = span\{\varphi_{ki}\}_{i\in\{1,\cdots,N\}}, \text{ for } \forall k \in \{1, \cdots, K\}, \quad (8)$$

where N is the number of local degrees of freedom.

Thus, the local solutions for y_h and $\boldsymbol{z}_h$ can be expressed in the form of local basis:

$$y_h(t, \boldsymbol{x})\Big|_{S_k} = \sum_{j=1}^{N} Y_{kj}(t)\varphi_{kj}(x), \boldsymbol{z}_h(t, \boldsymbol{x})\Big|_{S_k} = \sum_{j=1}^{N}\sum_{m=1}^{2} Z_{kj}(t)\boldsymbol{I}_m\varphi_{kj}(x), \quad (9)$$

where $\boldsymbol{I}_m$ represents the m-th unit vector in $\mathbb{R}^2$. We assume that there is a uniform polynomial degree p for every triangle.

Similarly, the coefficient functions are expressed by linear combinations of the basis functions on S_k, and we use the local representation vectors $[Y^0]_{k,:}$ for y_h^0, $[B]_{k,:}$ for b_h and $[F]_{k,:}$ for f_h.

Therefore, the semi-discrete formulation (7) with $\boldsymbol{w}_h = [\varphi_{ki}, 0]^T$ or $[0, \varphi_{ki}]^T$ and $v_h = \varphi_{ki}$ for $i \in 1, \cdots, N$ yields a time-dependent system whose contribution from S_k reads

$$\begin{aligned} &\sum_{j=1}^{N} Z_{kj}^m(t) \int_{S_k} \varphi_{ki}\varphi_{kj} - \sum_{j=1}^{N} Y_{kj}^m(t) \int_{S_k} \partial_{x^m}\varphi_{ki}\varphi_{kj} \\ &+ \int_{\partial S_k} \varphi_{k^-i}\boldsymbol{n}_{k^-}^m \widehat{y}_h = 0 \ (m = 1,\ 2), \\ &\sum_{j=1}^{N} \partial_t Y_{kj}(t) \int_{T_k} \varphi_{ki}\varphi_{kj} - \sum_{l=1}^{N} B_{kl}(t) \sum_{j=1}^{N}\sum_{m=1}^{2} Z_{kj}^m \int_{S_k} \partial_{x^m}\varphi_{ki}\varphi_{kl}\varphi_{kj} \\ &+ \int_{\partial S_k} \varphi_{k^-i}\widehat{\boldsymbol{z}}_h = \sum_{l=1}^{N} F_{kl}(t) \int_{S_k} \varphi_{ki}\varphi_{kl}, \end{aligned} \quad (10)$$

where $\widehat{y}_h$ and $\widehat{\boldsymbol{z}}_h$ are give as follows

$$\widehat{y}_h = \begin{cases} \dfrac{1}{2}\Big(\sum_{j=1}^{N} Y_{k^-j}\varphi_{k^-j} + \sum_{j=1}^{N} Y_{k^+j}\varphi_{k^+j}\Big) \text{ on } \varepsilon_\Omega, \\ y_D(t) \text{ on } \varepsilon_{\Omega_D}, \\ \sum_{j=1}^{N} Y_{k^-j}(t)\varphi_{k^-j} \text{ on } \varepsilon_{\Omega_N}, \end{cases} \quad (11)$$

and

$$\widehat{\boldsymbol{z}}_h = \begin{cases} \dfrac{1}{2}\sum_{m=1}^{2} n_{k^-}^m\Big(\sum_{l=1}^{N} B_{k^-l}(t)\varphi_{k^-l} \sum_{j=1}^{N} Z_{k^-j}^m\varphi_{k^-j} \\ \quad + \sum_{l=1}^{N} B_{k^+l}(t)\varphi_{k^+l} \sum_{j=1}^{N} Z_{k^+j}^m\varphi_{k^+j}\Big) \\ \quad + \dfrac{\eta}{h_{S_{k^-}}}\Big(\sum_{j=1}^{N} Y_{k^-j}\varphi_{k^-j} - \sum_{j=1}^{N} Y_{k^+j}\varphi_{k^+j}\Big) \text{ on } \varepsilon_\Omega, \\ \sum_{m=1}^{2} n_{k^-}^m \sum_{l=1}^{N} B_{k^-l}(t)\varphi_{k^-l} \sum_{j=1}^{N} Z_{k^-j}^m\varphi_{k^-j} \\ \quad + \dfrac{\eta}{h_{T_{k^-}}}\Big(\sum_{j=1}^{N} Y_{k^-j}\varphi_{k^-j} - y_D(t)\Big) \text{ on } \varepsilon_{\Omega_D}, \\ g_N(t) \sum_{l=1}^{N} B_{k^-l}(t)\varphi_{k^-l} \text{ on } \varepsilon_{\Omega_N}, \end{cases} \quad (12)$$

respectively, where the $\boldsymbol{n}_{S_k}$ is abbreviated by $\boldsymbol{n}_{S_k} = [n_k^1, n_k^2]$. The system (10) can be written in matrix form as

$$\begin{bmatrix} 0 \\ 0 \\ M\partial_t Y \end{bmatrix} + \begin{bmatrix} M & \mathbf{0} & -H^1 + Q^1 + Q_N^1 \\ \mathbf{0} & M & -H^2 + Q^2 + Q_N^2 \\ -G^1 + R^1 + R_D^1 & -G^1 + R^1 + R_D^1 & \eta(P + P_D) \end{bmatrix} \begin{bmatrix} Z^1 \\ Z^2 \\ C \end{bmatrix} = \begin{bmatrix} -J_D^1 \\ J_D^2 \\ \eta K_D - K_N + L \end{bmatrix}, \tag{13}$$

where the representation vectors $Z^m(t)$ ($m \in \{1, 2\}$) and $Y(t)$ are given as follows

$$Z^m(t) = \begin{bmatrix} Z_{11}^m & \cdots & Z_{1N}^m & \cdots & Z_{K1}^m & \cdots & Z_{KN}^m \end{bmatrix}^T,$$
$$Y^m(t) = \begin{bmatrix} Y_{11} & \cdots & Y_{1N} & \cdots & Y_{K1} & \cdots & Y_{KN} \end{bmatrix}^T.$$

The block matrices and the right-hand side vector of (13) are described as follows:

$$[M]_{(k-1)N+i,(k-1)N+j} = \int_{S_k} \varphi_{ki}\varphi_{kj} d\boldsymbol{x},$$

$$[H]^m_{(k-1)N+i,(k-1)N+j} = \int_{S_k} \partial_{x^m}\varphi_{ki}\varphi_{kj} d\boldsymbol{x},$$

$$[G]^m_{(k-1)N+i,(k-1)N+j} = \int_{S_k} \partial_{x^m}\varphi_{ki}\varphi_{kl}\varphi_{kj} d\boldsymbol{x},$$

$$L(t) = M\ [F_{11}(t)\cdots F_{1N}(t)\cdots\cdots F_{k1}(t)\cdots F_{KN}(t)]^T.$$

$$[Q^m]_{r,s} = \begin{cases} \dfrac{1}{2}\displaystyle\sum_{\xi_{kn}\in\partial_{T_k}\cap\varepsilon_\Omega} n_{kn}^m \int_{\xi_{kn}} \varphi_{ki}\varphi_{kj} \text{ for } r = s, \\ \dfrac{1}{2} n_{k^-n^-}^m \displaystyle\int_{\xi_{kn}} \varphi_{k^-i}\varphi_{k^+j} \text{ for } r \neq s, \end{cases}$$

$$[R^m]_{r,s} = \begin{cases} \dfrac{1}{2}\displaystyle\sum_{\xi_{kn}\in\partial_{T_k}\cap\varepsilon_\Omega} n_{kn}^m \sum_{l=1}^N B_{kl}(t) \int_{\xi_{kn}} \varphi_{ki}\varphi_{kl}\varphi_{kj} \text{ for } r = s, \\ \dfrac{1}{2} n_{k^-n^-}^m \displaystyle\sum_{l=1}^N B_{kl}(t) \int_{\xi_{kn}} \varphi_{k^-i}\varphi_{k^+l}\varphi_{k^+j} \text{ for } r \neq s, \end{cases}$$

$$[P]_{r,s} = \begin{cases} \dfrac{1}{|\xi_{kn}|}\displaystyle\int_{\xi_{kn}} \varphi_{ki}\varphi_{kj} \text{ for } r = s, \\ -\dfrac{1}{|\xi_{k^-n^-}|}\displaystyle\int_{\xi_{k^-n^-}} \varphi_{k^-i}\varphi_{k^+j} \text{ for } r \neq s, \end{cases}$$

$$[R_D^m]_{(k-1)N+i,(k-1)N+j} = \sum_{\xi_{kn}\in\partial_{S_k}\cap\varepsilon_D} n_{kn}^m \sum_{l=1}^N B_{kl}(t) \int_{\xi_{kn}} \varphi_{ki}\varphi_{kl}\varphi_{kj},$$

$$[P_D]_{(k-1)N+i,(k-1)N+j} = \sum_{\xi_{kn}\in\partial_{S_k}\cap\varepsilon_D} \frac{1}{|\xi_{kn}|} \int_{\xi_{kn}} \varphi_{ki}\varphi_{kj},$$

$$[Q_N^m]_{(k-1)N+i,(k-1)N+j} = \sum_{\xi_{kn}\in\partial_{S_k}\cap\varepsilon_N} n_{kn}^m \int_{\xi_{kn}} \varphi_{ki}\varphi_{kj},$$

$$[K_D]_{(k-1)N+i} = \sum_{\xi_{kn}\in\partial_{S_k}\cap\varepsilon_N} \frac{1}{|\xi_{kn}|} \int_{\xi_{kn}} \varphi_{ki} y_D(t),$$

$$[J_D^m]_{(k-1)N+i} = \sum_{\xi_{kn}\in\partial_{S_k}\cap\varepsilon_N} n_{kn}^m \int_{\xi_{kn}} \varphi_{ki} y_D(t),$$

$$[R_D^m]_{(k-1)N+i,(k-1)N+j} = \sum_{\xi_{kn}\in\partial_{S_k}\cap\varepsilon_N} \sum_{l=1}^N B_{kl}(t) \int_{\xi_{kn}} \varphi_{ki}\varphi_{kl} g_N(t).$$

B. Full-discrete scheme

We rewrite the semi-discrete scheme (13) as

$$W\partial_t C(t) + A(t)C(t) = V(t), \tag{14}$$

where $W \in \mathbb{R}^{3KN\times 3KN}$, $A(t) \in \mathbb{R}^{3KN\times 3KN}$, $V(t) \in \mathbb{R}^{3KN}$ and $C(t) \in \mathbb{R}^{3KN}$ are given as follows

$$W = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & M \end{bmatrix},$$

$$A = \begin{bmatrix} M & \mathbf{0} & -H^1 + Q^1 + Q_N^1 \\ \mathbf{0} & M & -H^2 + Q^2 + Q_N^2 \\ -G^1 + R^1 + R_D^1 & -G^1 + R^1 + R_D^1 & \eta(P + P_D) \end{bmatrix},$$

$$C(t) = \begin{bmatrix} Z^1(t) \\ Z^2(t) \\ Y(t) \end{bmatrix},\ V(t) = \begin{bmatrix} -J_D^1(t) \\ -J_D^2(t) \\ \eta K_D(t) - K_N(t) + L(t) \end{bmatrix},$$

respectively. Then, we use the second-order Crank-Nicholson method to discretize the system (13) in time. Let $0 = t^1 < \cdots < t^{N_t} = t_{end}$ be not a necessary uniform partition of the time interval T and let $\triangle t^n = t^{n+1} - t^n$ represent the time step size. So the CN-LDG full-discrete scheme is given as follows

$$W\frac{C^{n+1} + C^n}{2} = \frac{V^{n+1} + V^n}{2} - \frac{A^{n+1}C^{n+1} + A^nC^n}{2}, \tag{15}$$

where we abbreviate the $A^n = A(t^n)$, $C^n = C(t^n)$ and $V^n = V(t^n)$.

III. The reduced order model

The main idea of the reduced order CN-LDG method is the assumption that all solutions exist a low dimension manifold [10]. Thus, every numerical solution of FOM can be approximated as linear combination of a reduced number of global basis functions. In this paper, we introduce the POD method to construct a set of global POD basis functions and use it to generate the reduced order model of CN-LDG full-discrete scheme.

A. The construction of POD basis

First, we uniform extract the l provisional solutions $\{C^{n_i}\}_{n_i=0}^{l}$ $(0 \le n_1 \le \cdots \le l \ll N_t)$ of the CN-LDG scheme (15) as the snapshot vectors. It is worth noting that we only calculate the first l time steps of the CN-LDG scheme to generate snapshots in the numerical experiments. Furthermore, we obtain $3KN \times l$ snapshots matrix A_u

$$A_u = \begin{bmatrix} C_1^{n1} & C_1^{n2} & \cdots & C_1^l \\ C_2^{n1} & C_2^{n2} & \cdots & C_2^l \\ \vdots & \vdots & \ddots & \vdots \\ C_{3KN}^{n1} & C_{3KN}^{n2} & \cdots & C_{3KN}^l \end{bmatrix}, \tag{16}$$

where each column of snapshots matrix A_u is a snapshot vector. The singular value decomposition of the matrix A_u can be expressed as follows

$$A_u = L_u \begin{bmatrix} S^u_{r_u\times R_u} & \mathbf{0}_{r_u\times l-r_U} \\ \mathbf{0}_{(N-r_u)\times r_u} & \mathbf{0}_{(N-r_u)\times(l-r_u)} \end{bmatrix} R_u^T, \tag{17}$$

where L_u and R_u are $3KN \times 3KN$ and $l \times l$ unitary matrices, r_u is the rank of A_u, and $S^u_{r_u\times r_u} = diag(\sigma_{u,1}, \sigma_{u,2}, \cdots, \sigma_{u,r_u})$ with $\sigma_{u,1} \ge \sigma_{u,2} \ge \cdots \ge \sigma_{u,r_u} \ge 0$ is the singular values of A_u, where $L_u = (\psi_{u,1}, \psi_{u,2}, \cdots, \psi_{u,3KN})$ and $R_u = (\phi_{u,1}, \phi_{u,2}, \cdots, \phi_{u,l})$. The POD basis of dimension d_u $(d_u \le r_u)$ is given by the set $\{\psi_{u,i}\}_{i=1}^{d_u}$.

However, in practice, N may be extremely large, and it may not be possible to calculate the POD basis by SVD in matlab. In this case, we consider another efficient way to calculate the POD basis. From the SVD of the matrix A_u, we have

$$A_u\phi_{u,i} = \sigma_{u,i}\psi_{u,i}, \tag{18}$$

$$A_u^T\psi_{u,i} = \sigma_{u,i}\phi_{u,i}. \tag{19}$$

Talking (18) and (19) together, equations (20) and (21) are given as follows

$$A_u^T A_u\phi_{u,i} = \sigma_{u,i}A_u^T\psi_{u,i} = \sigma_{u,i}^2\phi_{u,i}, \tag{20}$$

$$A_u A_u^T\psi_{u,i} = \sigma_{u,i}A_u\phi_{u,i} = \sigma_{u,i}^2\psi_{u,i}, \tag{21}$$

where $i = 1, 2, \cdots, r_u$. According to (18)-(21), the POD basis functions can be obtained from the eigensystem of the correlation matrix $C_u = A_u^T A_u$ is given.

Proposition 3.1: Let $\lambda_{u,1} \geq \lambda_{u,2} \geq \lambda_{u,r_u} \geq 0$ represent the positive eigenvalues of the correlation matrix C_u and $\phi_{u,1}, \phi_{u,2}, \cdots, \phi_{u,r_u}$ denote the corresponding eigenvectors. Then, the POD basis are given as follows

$$\psi_{u,i} = \frac{1}{\sqrt{\lambda_{u,i}}}A_u\phi_{u,i} \ (i = 1, \cdots, d_u), \tag{22}$$

and the error bound is

$$\sum_{i=1}^{l}\left\|C^{n_i} - \sum_{j=1}^{d_u}(C^{n_i}, \psi_{u,j})\psi_{u,j}\right\|^2 = \sum_{j=d_u+1}^{r_u}\lambda_{u,j}. \tag{23}$$

Remark 3.2: The selection of the dimension d_u of the POD basis is not obvious. We hope the POD error bound to be less than some prescribed tolerance ρ. That is

$$\sum_{i=1}^{l}\left\|C^{n_i} - \sum_{j=1}^{d_u}(C^{n_i}, \psi_{u,j})\psi_{u,j}\right\|^2 \leq \rho. \tag{24}$$

Thus, d_u is selected to be the smallest integer such that

$$I(d_u) = \frac{\sum_{j=1}^{d_u}\lambda_{u,j}}{\sum_{j=1}^{r_u}\lambda_{u,j}} \geq 1 - \rho. \tag{25}$$

B. The reduced order model of CN-LDG scheme

The construction of the reduced order CN-LDG method is based on the application of the Galerkin projection on the broken polynomial space $\mathbb{P}_p(\mathcal{S}_h)$ and the POD basis space $V_{rb} = \{\psi_1, \psi_2, \cdots, \psi_{d_u}\}$. To describe the reduced order modeling process, we first introduce the Lemma 3.3.

Lemma 3.3: For the full rank matrix $A \in \mathbb{R}^{m\times n}$ and vector $b \in \mathbb{R}^n$, the projection coefficient x and projection matrix X of vector b on matrix A are given as follows

$$\begin{cases} x = (A^TA)^{-1}A^Tb, \\ X = A(A^TA)^{-1}A^T. \end{cases} \tag{26}$$

In the process of reduced order modeling, the reduced order solution C_r can be understood as the project of the high fidelity solution C on the POD basis space $V_{rb} = \{\psi_1, \psi_2, \cdots, \psi_{d_u}\}$. Based on Lemma 3.3, the numerical solution of ROM C_r and the project coefficient (reduced order coefficient) α_r are given as folows

$$C_r = \Psi_r(\Psi_r^T\Psi_r)^{-1}\Psi_r^TC = \Psi_r\Psi_r^TC, \tag{27}$$

$$\alpha_r = (\Psi_r^T\Psi_r)^{-1}\Psi_r^TC = \Psi_r^TC, \tag{28}$$

where each column of the matrix $\Psi_r = [\psi_1, \psi_2, \cdots, \psi_{d_u}]$ is made up of a POD basis. So we have

$$C(t) = \Psi_r\alpha_r(t). \tag{29}$$

TABLE I: The mesh sizes, discretization errors measured in L^2 and convergence orders for the LDG-$\mathbb{P}_1$ and LDG-$\mathbb{P}_2$ method.

	LDG-$\mathbb{P}_1$		LDG-$\mathbb{P}_2$	
Mesh sizes	L^2-error	order	L^2-error	order
0.5	1.23409e-1	-	4.13189e-2	-
0.25	3.44091e-2	1.84	7.03809e-3	2.55
0.125	8.9241e-3	1.94	8.62874e-4	3.02
0.0625	2.2492e-3	1.98	1.07557e-4	3.01

By substituting the above formula (29) into the spatial semi-discrete scheme (14), the residual vector $ResC(t)$ can be obtained

$$ResC(t) = W\Psi_r\partial_t\alpha_r(t) + A(t)\Psi_r\alpha_r(t) - V(t). \tag{30}$$

Then, using the Garlekin projection method, the following identity can be obtained

$$\Psi_r^T ResC(t) = \Psi_r^T W\Psi_r\partial_t\alpha_r(t) + \Psi_R^T A(t)\Psi_r\alpha_r(t) - \Psi_r^T V(t) \equiv 0. \tag{31}$$

So we get the global semi-discrete scheme of the ROM

$$\Psi_r^T W\Psi_r\partial_t\alpha_r(t) = \Psi_r^T V(t) - \Psi_r^T A(t)\Psi_r\alpha_r(t). \tag{32}$$

For the above equation (32), CN method is also used for time discretization, and the global fully discretizd ROM is given as follows

$$\begin{aligned} \Psi_r^T W\Psi_r\frac{\alpha_r^{n+1} + \alpha_r^n}{2} &= \frac{\Psi_r^T V^{n+1} + \Psi_r^T V^n}{2} \\ &- \frac{\Psi_r^T A^{n+1}\Psi_r\alpha_r^{n+1} + \Psi_r^T A^n\Psi_r\alpha_r^n}{2}. \end{aligned} \tag{33}$$

IV. Numerical experiments

In this section, we get some numerical results to certify the proposed CN-LDG-POD method. We consider the variable coefficients initial and boundary value diffusion problem (1) in a square domain $\Omega = \{(x_1, x_2) \in [0, 1] \times [0, 1]\}$ with Neumann boundary in $x_2 = 0, 1$ and Dirichlet boundaries elsewhere. The exact solution is given as follows

$$\begin{cases} y(t, \boldsymbol{x}) = \exp(-t)\cos(7x_1)\sin(7x_2), \\ b(t, \boldsymbol{x}) = t\exp(x_1 + x_2), \end{cases} \tag{34}$$

where the finite time interval is $t \in [0, \pi]$. The data y_D, g_N and f are derived algebraically by inserting y and b into (1). Firstly, we examine the convergence of LDG method and LDG-POD method. For the given time step $\triangle t$ satisfying the CFL type criterion, the convergence of the LDG method with the space of polynomial $\mathbb{P}_p(\mathcal{S}_h)$ $(p = 1, 2)$ is given in Table I. From the numerical results, the LDG method can reach the optimal convergence order for solving the variable coefficients diffusion equation. At same time, the convergence of the LDG-POD method with the space of polynomial $\mathbb{P}_p(\mathcal{S}_h)$ $(p = 1, 2)$ for solving the ROM is given in Table II. It still can reach the optimal convergence.

Then, In order to better reflect the superiority of the ROM in solving the variable coefficient diffusion equation, the

TABLE II: The mesh sizes, discretization errors measured in L^2 and convergence orders for the LDG-POD-$\mathbb{P}_1$ and LDG-POD-$\mathbb{P}_2$ method.

	LDG-POD-$\mathbb{P}_1$		LDG-POD-$\mathbb{P}_2$	
Mesh sizes	L^2-error	order	L^2-error	order
0.5	1.23408e-1	-	4.13197e-2	-
0.25	3.44090e-2	1.84	7.03783e-3	2.54
0.125	8.92395e-3	1.94	8.62839e-4	3.01
0.0625	2.24896e-3	1.98	1.14514e-4	2.91

detailed computational performance is given in Table III. Here, we take the dimension of reduced order basis functions $d_u = 4$. It should be noted that the L^2-error in the table is the relative error, which is the L^2-error between the numerical solutions of the CN-LDG and CN-LDG-POD methods. In the FOM, the CPU time is composed of T_{fc} and T_{fs}, where T_{fc} represents the time taken to build the global system, and T_{fs} is the time taken to solve the CN-LDG fully discrete scheme. However, the CPU time is divided into T_{rc}, T_{rs} and T_{POD} in the ROM, where $T_{rc} = T_{fc}$, T_{rs} represents the time taken to solving the CN-LDG-POD fully discrete scheme (32), and T_{POD} is the time taken to build the POD basis. We define T_{rate} as follows

$$T_{rate} = \frac{T_{fs}}{T_{rs} + T_{POD}}, \tag{35}$$

it is the computational time saving rate of the ROM. We can see from the Table III, where N_{dof}^{FOM} and N_{dof}^{ROM} represent the degrees of freedom of FOM and ROM, respectively. Obviously, from the Table III, the CN-LDG-POD method can effectively decrease the computational time, while maintain the high precision of the original algorithm.

TABLE III: Comparisons between the FOM and ROM in terms of CPU time and L^2-error.

			FOM	ROM				
P_i	$T_{fc}(=T_{rc})$	N_{dof}^{FOM}	T_{fs}	N_{dof}^{ROM}	T_{rs}	T_{POD}	T_{rate}	$\|Er\|_{L^2}$
1	399.7	18432	74.86	4	12.57	8.04	27.54%	7.11818e-6
2	547.6	36864	286.8	7	44.46	21.47	22.29%	3.02127e-6

V. Conclusion and outlook

In this paper, we present a model reduction method based on POD method for the diffusion equation with variable coefficients. We use the CN-LDG method to construct the FOM. The POD method and Galerkin projection method are used to construct the ROM of CN-LDG method. Compared with the FOM, the degree of freedom of the ROM is greatly reduced, so it can effectively reduce the computational cost. At the same time, the ROM can maintain the numerical accuracy of the FOM. We give some numerical results to prove the effectiveness of the proposed order reduction method. Next, we consider apply the presented model reduction method to solve more complex models, such as fluid-fluid coupling, ferrofluid and porous media models.

References

[1] V. Aizinger, C. Dawson, B. Cockburn, P. Castillo, The local discontinuous Galerkin method for contaminant transport, Adv. Water Resour. 24 (1) (2000) 73-87.

[2] J. Burkardt, M. Gunzburger, H. C. Lee, POD and CVT-based reduced-order modeling of Navier-Stokes flows, Comput. Methods Appl. Mech. Eng. 196 (1-3) (2006) 337-355.

[3] P. H. Cai, L. Z. Qian, Implicit-explicit time stepping scheme based on the streamline diffusion method for fluid-fluid interaction problems, IAENG International Journal of Applied Mathematics. 48 (3) (2018) 278-287.

[4] B. Cockburn, C. W. Shu, Tvb runge-kutta local projection discontinuous galerkin finite element method for conservation laws. ii. general framework, Math. Comput. 84 (1) (1989) 90-113.

[5] B. Cockburn, C. W. Shu, The local discontinuous galerkin method for time-dependent convection-diffusion systems, SIMA J. Numer. Anal. 35 (6) (1998) 2440-2463.

[6] K. Carlberg, C. Farhat, J. Cortial, D. Amsallem, The GNAT method for nonlinear model reduction: Effective implementation and application to computationalfluid dynamics and turbulent flows, J. Comput. Phys. 242 (2013) 623-647.

[7] F. Frank, B. Reuter, V. Aizinger, P. Knaber, FESTUNG: A MATLAB/GUN Octave toolbox for the discontinuous Galerkin method, Part I: Diffusion Operator, Comput. Math. Appl. 70 (1) (2015) 11-46.

[8] H. Fahs, S. Lanteri, A high-order non-conforming discontinuous Galerkin method for time-domain electromagnetics, J. Comput. Appl. Math. 234 (4) (2010) 1088-1096.

[9] K. Li, T. Z. Huang, L. Li, et al. A reduced-order discontinuous Galerkin method based on POD for electromagnetic simulation, IEEE Trans. Antennas Propag. 66 (1) (2018) 242-254.

[10] K. Li, T. Z. Huang, L. Li, et al. A reduced-order dicontinuos Galerkin method based on a Krylov subspae techniquein nanophotonics, Comput. Math. Appl. 358 (2019) 128-145.

[11] L. Li, S. Lanteri, R. Perrussel, Numerical investigation of a high order hybridizable discontinuous Galerkin method for 2D time-harmonic Maxwells equations, COMPEL-Int. J. Comput. Math. Elect. Electron. Eng. 32 (3) (2013) 1112-1138.

[12] N. Ploymaklam, S. Chaturantabut, Reduced-order modeling of a local discontinuous Galerkin method for Burgers-Poisson equations, Thai J. Math. 18 (4) (2020) 2053-2069.

[13] H. Reed, T. R. Hill, Triangular mesh methods for the Neutron transport equation, Tech. Rep. LA-UR-73-479. Los alamos report La. (1973).

[14] S. M. Schnepp, T. Weiland, Efficient large scale electromagnetic simulations using dynamically adapted meshes with the discontinuous Galerkin method, J. Comput. Appl. Math. 236 (18) (2012) 4909-4924.

[15] L. Sirovich, Turbulence and the dynamics of coherent structures part I: Coherent structures, Quart. Appl. Math. 45 (3) (1986) 561-571.

[16] X. Ye, P. Li, M. Zhao, R. Panda, J. Hu, Scalable analysis of mesh-based clock distribution networks using application-specific reduced order modeling, IEEE. T. Comput. Aid. D. 29 (2010) 1342-1353.

Dan-chen Zhu was born in Fengcheng, Jiangxi Province, china, in 1999. The author is a graduate student in computational mathematics at Guangxi Normal University, Guilin, Guangxi Zhuang Autonomous Region, China, starting in 2022.

The author current research interests include numerical solution of partial differential equations and computing sciences, two-phase fluids, discontinuous Galerkin finite element method.

Ling-zhi Qian was born in Lingbi, Anhui Province, china, in 1980. The author received his Ph. D. in computational mathematics at Nanjing Normal University, Nanjing, Jiangsu Province, China, in June 2016.

The author current research interests include numerical solution of partial differential equations and computing sciences, fluids and fluid-fluid interaction problems, two-phase fluids.

Jing Wang was born in Shaoyang, Hunan Province, china, in 1998. The author is a graduate student in computational mathematics at Guangxi Normal University, Guilin, Guangxi Zhuang Autonomous Region, China, starting in 2020.

The author current research interests include numerical solution of partial differential equations and computing sciences, fluids and fluid-fluid interaction problems, discontinuous Galerkin finite element method.