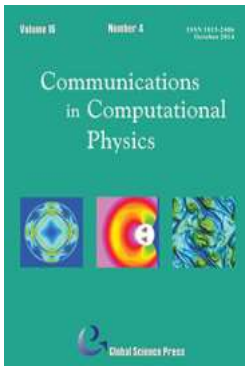


Fast Solver for the Local Discontinuous Galerkin Discretization of the KdV Type Equations

Ruihan Guo and Yan Xu

Communications in Computational Physics / Volume 17 / Issue 02 / February 2015, pp 424 - 457
DOI: 10.4208/cicp.210114.080814a, Published online: 22 January 2015

Commun. Comput. Phys.
doi: 10.4208/cicp.210114.080814a

Vol. **17**, No. 2, pp. 424-457
February 2015

Fast Solver for the Local Discontinuous Galerkin Discretization of the KdV Type Equations

Ruihan Guo[1] and Yan Xu[1,*]

[1] *School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui 230026, P.R. China.*

Received 21 January 2014; Accepted (in revised version) 8 August 2014

Communicated by Chi-Wang Shu

Abstract. In this paper, we will develop a fast iterative solver for the system of linear equations arising from the local discontinuous Galerkin (LDG) spatial discretization and additive Runge-Kutta (ARK) time marching method for the KdV type equations. Being implicit in time, the severe time step ($\Delta t = \mathcal{O}(\Delta x^k)$, with the k-th order of the partial differential equations (PDEs)) restriction for explicit methods will be removed. The equations at the implicit time level are linear and we demonstrate an efficient, practical multigrid (MG) method for solving the equations. In particular, we numerically show the optimal or sub-optimal complexity of the MG solver and a two-level local mode analysis is used to analyze the convergence behavior of the MG method. Numerical results for one-dimensional, two-dimensional and three-dimensional cases are given to illustrate the efficiency and capability of the LDG method coupled with the multigrid method for solving the KdV type equations.

AMS subject classifications: 65M60, 35Q53

Key words: KdV type equations, local discontinuous Galerkin methods, multigrid algorithm, additive Runge-Kutta methods, local mode analysis.

1 Introduction

In this paper, we apply the multigrid (MG) solver to solve the system of algebraic equations arising from the local discontinuous Galerkin (LDG) spatial discretization and additive Runge-Kutta (ARK) time marching method for the KdV type equations containing third derivatives terms

$$u_t + f(u)_x + u_{xxx} = 0, \tag{1.1}$$

*Corresponding author. *Email addresses:* guoguo88@mail.ustc.edu.cn (R. Guo), yxu@ustc.edu.cn (Y. Xu)

and the fifth-order KdV type equations

$$u_t+f(u)_x+g(u_x)_{xx}+u_{xxxxx}=0, \tag{1.2}$$

in $\Omega\in\mathbb{R}^d(d\leq 3)$, where $f(u)\geq 0$ and $g(p)$ are arbitrary (smooth) functions.

The LDG methods for these two types of equations were derived by Yan and Shu [18, 19], which were high order accurate, stable and flexible for arbitrary h and p adaptivity. In these two papers, time discretization was by the explicit Runge-Kutta method with a suitably small Δt for stability ($\Delta t=\mathcal{O}(\Delta x^3)$ for Eq. (1.1) and $\Delta t=\mathcal{O}(\Delta x^5)$ for Eq. (1.2)). Usually, it is not necessary to choose such a small time step for the purpose of accuracy and is purely an artifact of the explicit time discretization technique. Therefore, implicit methods should be used to improve the computational efficiency.

The discontinuous Galerkin (DG) method is a class of finite element methods using completely discontinuous basis functions, which are usually chosen as piecewise polynomials. Reed and Hill [10] first introduced the DG method in 1973, in the framework of neutron linear transport. For PDEs containing higher order spatial derivatives, the DG method can also be applied directly, Liu and Yan [9] developed direct DG methods for diffusion problems. Then, Bona et al. constructed conservative DG methods for the general KdV equation in [3]. The first LDG method was constructed by Cockburn and Shu in [4] as an extension of the Runge-Kutta DG method to general convection-diffusion problems. The idea of the LDG method is to rewrite the equations with higher order derivatives as a first order system, then apply the DG method to the system. The LDG techniques have been developed for various high order PDEs including the convection diffusion equations [4], nonlinear one-dimensional and two-dimensional KdV type equations [16,18]. More details about the LDG methods for high-order time dependent PDEs can be found in the review paper [17].

Xia et al. [14] explored the ARK method to solve the stiff ordinary differential equations (ODEs) resulting from an LDG spatial discretization to PDEs with higher order spatial derivatives and found that it was an efficient time discretization method. The implicit method requires to solve system of linear equations at each time step. The efficiency of the method highly depends on the efficiency of the solver for the linear systems. In [14], the resulting linear system of algebraic equations were solved by direct linear solver in LAPACK, which was not efficient for high-dimensional problems. Other traditional iterative methods such as Gauss-Seidel method suffer from slow convergence rates for large scale problems. Thus, we devote to developing an iterative fast solver for the system of linear equations.

The multigrid (MG) method was originally applied to simple boundary value problems, e.g. second-order boundary value problem. Then, the MG method was extended to solve time-dependent PDEs with even-order spatial derivatives and found that it was an efficient method. Recently, the MG method coupled with the DG spatial discretization for the compressible Naiver-Stokes equation [7,11] and the Euler equation [1,2] had been studied. In [12,13], the MG method was introduced to solve the system of algebraic equations arising from the higher order DG discretization of advection dominated flows. Guo

and Xu studied the multigrid solver coupled with the LDG method for the Cahn-Hilliard equation in [5].

The KdV type equations (1.1) and (1.2) are time-dependent PDEs with odd-order spatial derivatives. The use of LDG spatial discretization and implicit time marching method for Eqs. (1.1) and (1.2) will typically result in linear system of algebraic equations at each time step, which are non-symmetric systems. In this paper, we will apply the MG method to solve the systems and we numerically show the optimal or sub-optimal complexity of the MG solver. Thus, we have a conclusion that the MG method is efficient for algebraic equations arising by implicit time integration methods and LDG spatial discretizations for PDEs with odd-order spatial derivatives.

In order to predict the MG behavior, a two-level local mode analysis is used to study the convergence of the MG method. Although we restrict ourselves to one-dimensional problems, with considerably extra complexity, a similar analysis can be made for two- or three- dimensional problems by the tensor product principle.

The paper is organized as follows. In Section 2, we present an example to indicate that the MG method is efficient for the KdV type equations. The LDG method and the two-level local mode analysis for Eq. (1.1) are presented in Section 3, while for Eq. (1.2), the same information is shown in Section 4. In Section 5, we extend the LDG method and MG solver to the general odd-order linear PDEs. Section 6 contains numerical results for the KdV type equations for one-dimensional, two-dimensional and three-dimensional cases. Finally we give concluding remarks in Section 7.

2 Motivation

The MG method is mainly used to solve PDEs with even-order spatial derivatives. There is little work for PDEs with odd-order spatial derivatives and what we want to know is that, does the MG method work for this type of equations? To answer the question, we will first give an example to illustrate the efficiency of the MG method for the KdV type equations containing third derivatives terms.

Example 2.1. Consider one-dimensional linear KdV equation

$$u_t + u_{xxx} = 0, \tag{2.1}$$

with periodic boundary conditions in $\Omega=(0,2\pi)$. We apply the LDG spatial discretization (for a detailed description of the LDG method for Eq. (2.1), we refer the readers to [18]) and backward Euler time marching method to Eq. (2.1) with $\Delta t = 0.1\Delta x$, then it requires to solve a system of linear equations at each time step. The damped Jacobi (damping parameter $\alpha=0.7$) method and the two-grid correction method (for a detailed description of the method, we refer the readers to [5]) are applied to solve the linear system, on a grid with $N=64$ points, respectively. We use an initial guess

$$u_j^h = \frac{1}{2}\left[\sin\left(\frac{16j\pi}{N}\right)+\sin\left(\frac{40j\pi}{N}\right)\right],$$

consisting of the $k=16$ and $k=40$ modes.

The results of this calculation are given in Fig. 1. In the top left, The approximation u^h after one relaxation sweep on the fine grid is superimposed on the initial guess (the dotted line). The top right, middle left and middle right plots show the approximation after three, four and five relaxation sweeps, respectively on the fine grid, superimposed on the initial guess. Much of the oscillatory component of the initial guess has already been moved. Further relaxations on the fine grid would provide only a slow improvement at this point. Thus, it is time to move to the coarse grid.

The bottom left shows the fine-grid error after three relaxation sweeps on the coarse-grid residual equation, superimposed on the initial guess. The error is reduced by moving to the coarse grid. After three additional fine-grid relaxations, the error is reduced again and the result is plotted in the bottom right figure.

From Fig. 1, we can see that the MG method is efficient to solve the linear system of equations arising from the LDG spatial discretization and backward Euler time marching method for the KdV type equations. Thus, we will develop a high order time integration method and apply the MG method to solve the equations arising by the KdV type equations (1.1) and (1.2) in the next sections.

3 The KdV type equations

3.1 Notations

Let $\mathcal{T}_h$ denote a tessellation of Ω with shape-regular element K. Let Γ denote the union of the boundary faces of elements $K\in\mathcal{T}_h$, i.e. $\Gamma=\bigcup_{K\in\mathcal{T}_h}\partial K$, and $\Gamma_0=\Gamma\backslash\partial\Omega$.

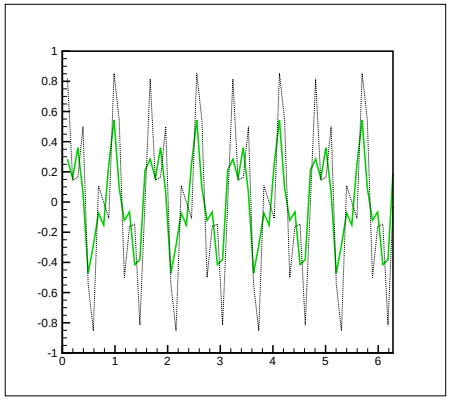
In order to describe the flux functions, we need to introduce some notations. Let e be a face shared by the "left" and "right" elements K_L and K_R (we refer to [17] for more details of the definition). Define the normal vectors $\boldsymbol{\nu}_L$ and $\boldsymbol{\nu}_R$ on e pointing exterior to K_L and K_R, respectively. If ψ is a function on K_L and K_R, but possibly discontinuous across e, let ψ_L denote $(\psi|_{K_L})|_e$ and ψ_R denote $(\psi|_{K_R})|_e$, the left and right trace, respectively.

Let $\mathcal{P}^k(K)$ be the space of polynomials of degree at most $k\geq 0$ on K. The finite element spaces associated with the mesh are of the form

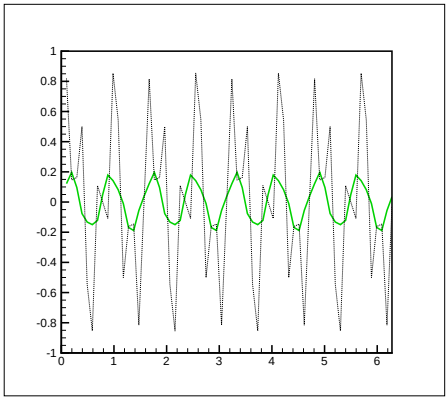
$$V_h^k=\{v\in L^2(\Omega):v|_K\in\mathcal{P}^k(K),\forall K\in\mathcal{T}_h\},$$
$$\Sigma_h^k=\{\boldsymbol{w}=(w_1,\cdots,w_d)^T\in L^2(\Omega)^d:w_l|_K\in\mathcal{P}^k(K),l=1,\cdots,d,\forall K\in\mathcal{T}_h\}.$$

Further, we define the inner product notations as

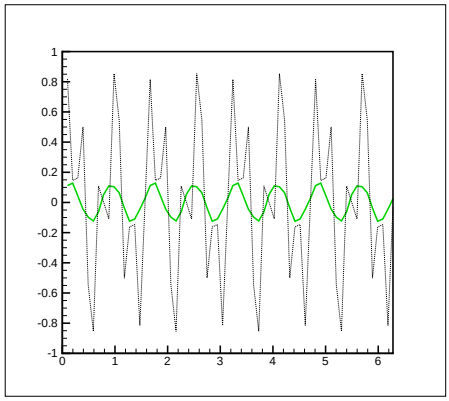
$$(w,v)_K=\int_K wvdK,\quad (w,v)_{\partial K}=\int_{\partial K}wvds, \tag{3.1}$$
$$(\boldsymbol{q},\boldsymbol{p})_K=\int_K \boldsymbol{q}\cdot\boldsymbol{p}dK,\quad (\boldsymbol{q},\boldsymbol{p})_{\partial K}=\int_{\partial K}\boldsymbol{q}\cdot\boldsymbol{p}ds, \tag{3.2}$$



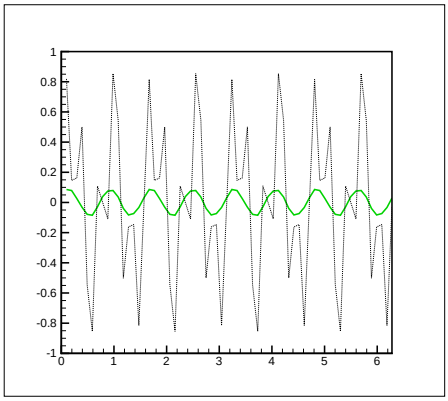
(a) The error after one sweep of weighted Jacobi.



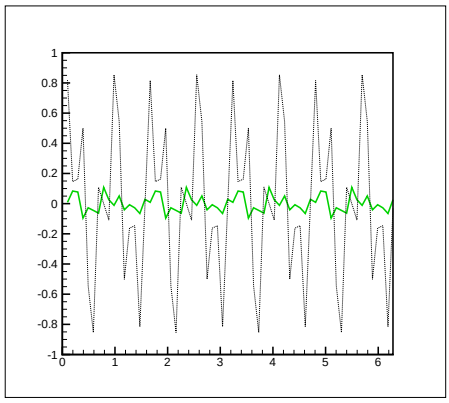
(b) The error after three sweeps of weighted Jacobi.



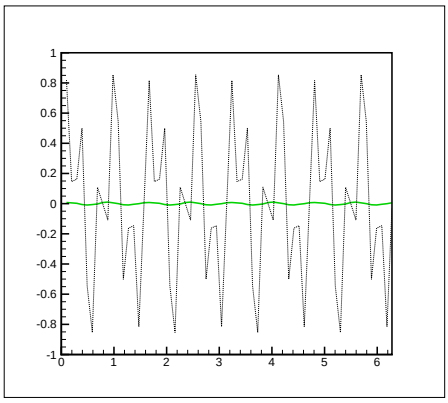
(c) The error after four sweeps of weighted Jacobi.



(d) The error after five sweeps of weighted Jacobi.



(e) The fine-grid error after three sweeps of weighted Jacobi on the coarse-grid problems.



(f) The fine-grid error after the coarse-grid correction is followed by three weighted Jacobi sweeps on the fine grid.

Figure 1: Coarse-grid correction for Eq. (2.1) on a grid with $N=64$.

for the scalar variables w, v and vector variables $\boldsymbol{q}$, $\boldsymbol{p}$ respectively. The inner products on Ω are defined as

$$(w,v)_\Omega=\sum_K(w,v)_K,\quad (\boldsymbol{q},\boldsymbol{p})_\Omega=\sum_K(\boldsymbol{q},\boldsymbol{p})_K. \tag{3.3}$$

Here we only consider periodic boundary conditions. Notice that the assumption of periodic boundary conditions is for simplicity and not essential: the method can be easily designed for non-periodic boundary conditions. The development of the LDG method for the non-periodic boundary conditions can be found in [8].

3.2 The LDG scheme for the KdV type equations

In this section, we apply the LDG spatial discretization to the KdV type equations (1.1). To ease the presentation, we consider the simpler one-dimensional linear problems

$$u_t+u_{xxx}=0. \tag{3.4}$$

To define the LDG method to Eq. (3.4), we rewrite it as a first order system:

$$u_t+p_x=0,\quad p-q_x=0,\quad q-u_x=0. \tag{3.5}$$

To simplify the notation, we still use u, p and q to denote the numerical solution. Applying the DG method to the system (3.5), we have the scheme: Find $u,p,q\in V_h^k$, such that, for all test functions $\phi,\theta,\eta\in V_h^k$, we have

$$(u_t,\phi)_K-(p,\phi_x)_K+(\hat{p},\phi)_{\partial K}=0, \tag{3.6a}$$
$$(p,\theta)_K+(q,\theta_x)_K-(\hat{q},\theta)_{\partial K}=0, \tag{3.6b}$$
$$(q,\eta)_K+(u,\eta_x)_K-(\hat{u},\eta)_{\partial K}=0. \tag{3.6c}$$

The "hat" terms in (3.6) at the cell boundary from integration by parts are the so-called "numerical fluxes", which are functions defined on the edges and should be designed based on different guiding principles for different PDEs to ensure stability. Following the discussion in [16,18,19], we can take the simple choices such that

$$\hat{p}|_e=p_R,\quad \hat{q}|_e=q_R,\quad \hat{u}|_e=u_L. \tag{3.7}$$

Considering the scheme (3.6), we aim to eliminate the auxiliary variables, obtain an ODE and apply the backward Euler time marching method. The energy stability for this fully discretized scheme was proved in [18]. In the following, we restrict ourselves to the one-dimensional equation because this is the building block for the high-dimensional case. For space discretization, we assume uniform meshes. In one-dimension space, the computational domain can be written as $\Omega=\bigcup_{j=1}^N I_j$ and $I_j=[x_{j-\frac{1}{2}},x_{j+\frac{1}{2}}]$. We take $\mathcal{P}^1$

approximation for example and take the basis function $\{\phi_i\}_{i=0}^1$, the approximate solution reads

$$u=\sum_{j=1}^{N}\sum_{i=0}^{1}u_j^i\phi_i(2(x-x_j)/\Delta x),$$

and at each cell I_j, according to the scheme (3.6a), (3.6b) and (3.6c), with the choice of flux (3.7), the coefficients $\boldsymbol{u}_j=[u_j^0,u_j^1]^T$, $\boldsymbol{p}_j=[p_j^0,p_j^1]^T$ and $\boldsymbol{q}_j=[q_j^0,q_j^1]^T$ satisfy

$$\frac{1}{\Delta t}\boldsymbol{u}_j^n-\frac{1}{\Delta t}\boldsymbol{u}_j^{n+1}=M_1\boldsymbol{p}_j^{n+1}+M_R\boldsymbol{p}_{j+1}^{n+1}, \tag{3.8a}$$

$$\boldsymbol{p}_j^{n+1}=M_1\boldsymbol{q}_j^{n+1}+M_R\boldsymbol{q}_{j+1}^{n+1}, \tag{3.8b}$$

$$\boldsymbol{q}_j^{n+1}=M_2\boldsymbol{u}_j^{n+1}+M_L\boldsymbol{u}_{j-1}^{n+1}, \tag{3.8c}$$

respectively, where

$$M_1=\frac{1}{\Delta x}\begin{bmatrix}-1 & 1\\ -3 & -3\end{bmatrix},\quad M_R=\frac{1}{\Delta x}\begin{bmatrix}1 & -1\\ 3 & -3\end{bmatrix},$$
$$M_2=\frac{1}{\Delta x}\begin{bmatrix}1 & 1\\ -3 & 3\end{bmatrix},\quad M_L=\frac{1}{\Delta x}\begin{bmatrix}-1 & -1\\ 3 & 3\end{bmatrix}.$$

There are two methods to eliminate the auxiliary variables, i.e. we have two methods to obtain $\boldsymbol{u}^{n+1}$ from the known $\boldsymbol{u}^n$.

- **Method 3X**
  We eliminate $\boldsymbol{q}_j^{n+1}$ and $\boldsymbol{p}_j^{n+1}$ from Eqs. (3.8c) and (3.8b), respectively. Then we get a linear equation

  $$K\boldsymbol{u}^{n+1}=\boldsymbol{f}, \tag{3.9}$$

  where K is described by the repetition of stencil

  $$\begin{aligned}&[-\Delta tM_1^2M_L|-I-\Delta t(M_1^2M_2+M_1M_RM_L+M_RM_1M_L)|\\ &\quad-\Delta t(M_1M_RM_2+M_RM_1M_2+M_R^2M_L)|-\Delta tM_R^2M_2],\end{aligned}$$

  $\boldsymbol{f}$ is the corresponding right hand side vector consisting of $\boldsymbol{u}^n$ and I is the identical matrix.

- **Method X-2X**
  We only eliminate $\boldsymbol{q}_j^{n+1}$ from Eq. (3.8c), then we get a linear equation with double degree of freedom comparing to Eq. (3.9) and it takes the form

  $$G\boldsymbol{U}=\boldsymbol{F}, \tag{3.10}$$

where $\boldsymbol{U}=[[u_1^0,u_1^1,p_1^0,p_1^1,\cdots,u_j^0,u_j^1,p_j^0,p_j^1,\cdots,u_N^0,u_N^1,p_N^0,p_N^1]^{n+1}]^T$ and G is described by the repetition of stencil

$$\left[\left[\begin{array}{cc}\mathbf{0} & \mathbf{0}\\ -M_1M_2 & \mathbf{0}\end{array}\right]\left|\left[\begin{array}{cc}-I & -\Delta tM_1\\ -(M_1M_2+M_RM_L) & I\end{array}\right]\right|\left[\begin{array}{cc}\mathbf{0} & -\Delta tM_R\\ -M_RM_2 & \mathbf{0}\end{array}\right]\right]$$

and $\boldsymbol{F}$ is the corresponding right hand side vector consisting of $\boldsymbol{u}^n$ and $\boldsymbol{p}^n$.

Remark 3.1. We will show the convergence behavior of the two methods, which indicate Method X-2X has better convergence behavior than Method 3X. The main reason is that the condition number of the discretization matrix G is smaller than the discretization matrix K. But Method X-2X will require more computational memory cost than Method 3X.

Remark 3.2. The LDG scheme for more general KdV type equations in multi-dimensional case can be found in [16,18,19]. The treatment for the linear system is similar. We will also show the numerical results for more general KdV type equations in multi-dimensional case in Section 6.

3.3 Numerical investigation of the convergence behavior for the multigrid method

Now the linear system is obtained by the LDG spatial discretization and backward Euler time marching method and we are ready to introduce an efficient solver to solve the linear system of algebraic equations (3.9) and (3.10). Traditional iterative methods such as Gauss-Seidel method suffer from slow convergence rates for large scale problems. From Section 2, we find that the MG method is efficient to solve the linear system of equations derived by the LDG spatial discretization and backward Euler time marching method for the KdV type equations. Thus, we will apply the MG method to solve Eqs. (3.9) and (3.10).

At the core of any MG method is the two-grid cycle, which we consider first. A fairly complete description of the two-grid cycle algorithm that is used to solve the linear equations at the implicit time level is given in [5]. As to the convergence behavior of the two-grid algorithm, we apply the local mode analysis to the error amplification operator $E_h^{2\text{grid}}$ to compute the eigenvalue spectra of the operator. For a detailed description of the local mode analysis, we refer the readers to [5].

We take Method 3X for example, the amplification operator E_h is defined as

$$E_h=I-S_hA_h, \tag{3.11}$$

where $A_h=K$ has been defined, S_h is the smoother and I is the identity operator in V_h^k. The amplification operator of the two-grid algorithm is given by

$$E_h^{2\text{grid}}=E_h^{\nu_2}[I-P_{hH}A_H^{-1}R_{Hh}A_h]E_h^{\nu_1}, \tag{3.12}$$

where ν_1 and ν_2 are the number of pre-(post-) relaxation sweeps, respectively, and A_H, P_{hH}, R_{Hh} are defined in [5]. While for Method X-2X, A_h is G and we can get E_h and $E_h^{2\text{grid}}$. The spectral radius, or some norm (some norm) of this operator allows to quantify how the error is reduced at each iteration. If it is less than 1, we will get a convergent iteration. The smaller it is, the faster is the iteration.

The eigenvalue spectra of E_h and $E_h^{2\text{grid}}$ for Method 3X and Method X-2X are shown in Figs. 2-5. The damping parameter α is chosen by the figures which present asymptotic convergence factor λ changes with α. From these figures, we have

- The eigenvalue spectra of E_h is near or larger than 1 and the eigenvalue spectra of E_h^{2grid} is strictly less than 1, i.e. the damped Gauss-Seidel and Jacobi iteration methods are not convergent, while the two-grid algorithm with damped Gauss-Seidel or Jacobi smoother is convergent.
- The convergence behavior for damped Gauss-Seidel smoother is similar to damped Jacobi smoother.
- Method X-2X has better convergence behavior than Method 3X.

3.4 The additive Runge-Kutta method

Xia et al. [14] explored the ARK method and found that it was an efficient high order time marching method when it was coupled with the LDG spatial discretization. In this paper, we are devoted to obtaining a high order scheme both in time and space. Thus the ARK method is applied to solve the ODEs, which is obtained by the LDG spatial discretization operator. Take Method 3X for example, we choose a local basis in cell K, then p and q can be eliminated from Eqs. (3.6b) and (3.6c), respectively, by simply inverting a small mass matrix in each case, and it yields an ODE

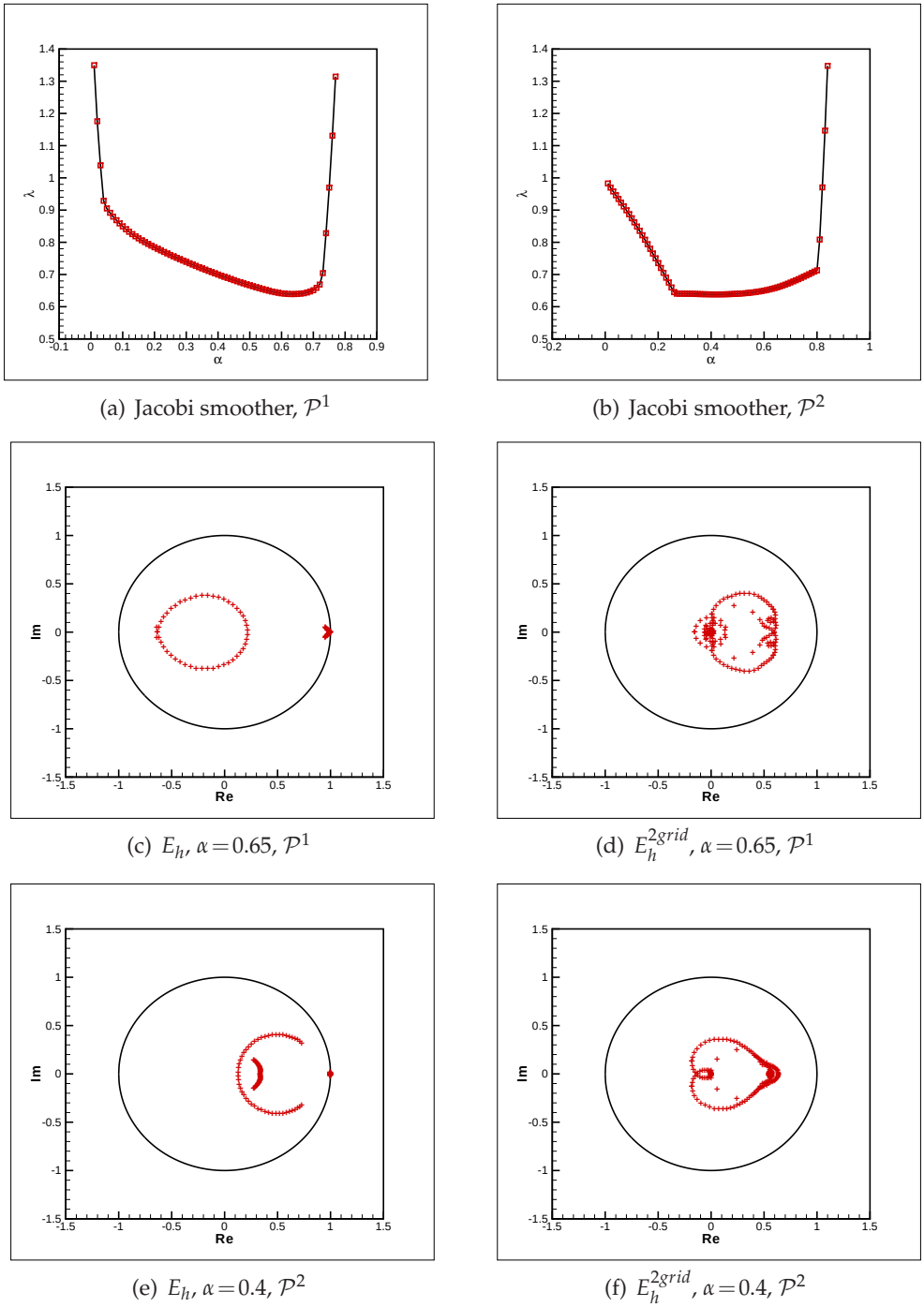
$$\boldsymbol{u}_t = L(\boldsymbol{u}). \tag{3.13}$$

Thus, the ARK scheme, which is used to solve the ODE (3.13), is given in the following form

$$\boldsymbol{u}^{(i)} = \boldsymbol{u}^n + \Delta t \sum_{j=0}^{i} a_{ij} L(t^n + c_j \Delta t, \boldsymbol{u}^{(j)}), \quad i = 1, \cdots, s \tag{3.14a}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \Delta t \sum_{j=0}^{s} b_j L(t^n + c_j \Delta t, \boldsymbol{u}^{(j)}), \tag{3.14b}$$

where $\boldsymbol{u}^{(0)} = \boldsymbol{u}^n$ and $\boldsymbol{u}^{(i)}$ approximates $\boldsymbol{u}(t^n + c_i \Delta t)$. The Butcher coefficients a_{ij}, b_j and c_j are constrained by order of accuracy and stability considerations. In [6], the implicit-explicit ARK methods from third- to fifth-order are presented in which the stiff terms

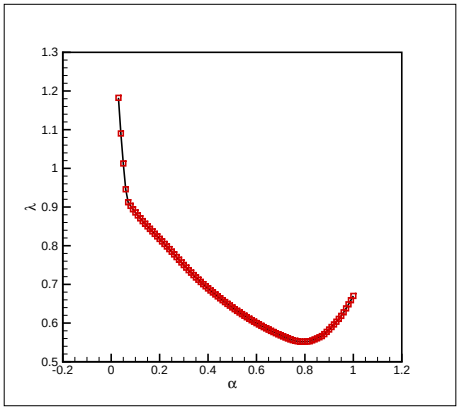


(a) Jacobi smoother, $\mathcal{P}^1$ (b) Jacobi smoother, $\mathcal{P}^2$

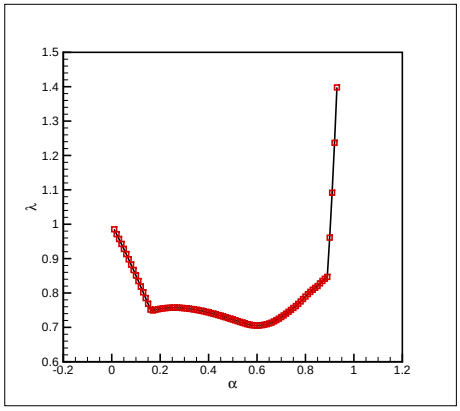
(c) E_h, $\alpha=0.65$, $\mathcal{P}^1$ (d) E_h^{2grid}, $\alpha=0.65$, $\mathcal{P}^1$

(e) E_h, $\alpha=0.4$, $\mathcal{P}^2$ (f) E_h^{2grid}, $\alpha=0.4$, $\mathcal{P}^2$

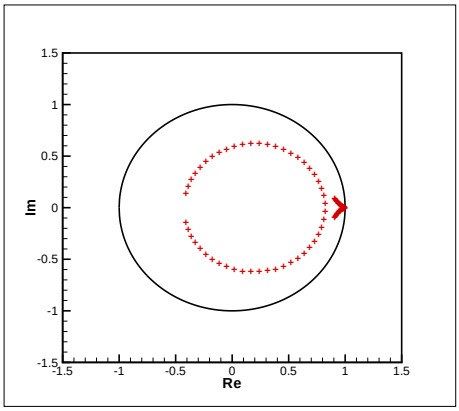
Figure 2: Method 3X, cell number $N=128$, $\Delta t=0.1\Delta x$, damped Jacobi smoother, $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation. Figs. 2(a)-2(b) show the asymptotic convergence factor λ changes with the damping parameter α. Figs. 2(c)-2(f) show the eigenvalue spectra of E_h and E_h^{2grid}.



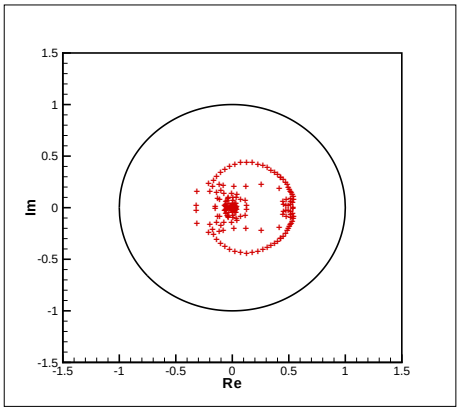
(a) Gauss-Seidel smoother, $\mathcal{P}^1$



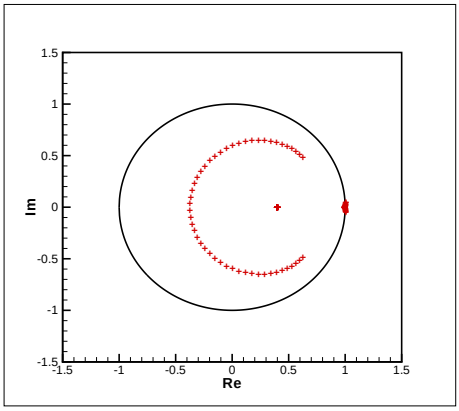
(b) Gauss-Seidel smoother, $\mathcal{P}^2$



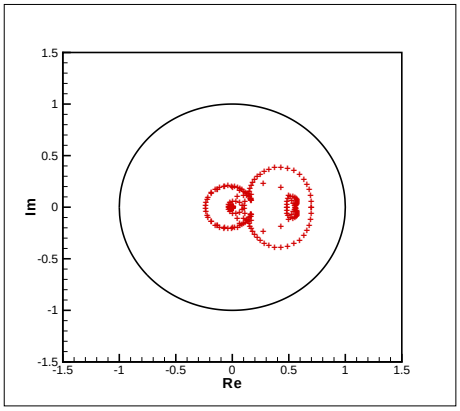
(c) E_h, $\alpha=0.8$, $\mathcal{P}^1$



(d) E_h^{2grid}, $\alpha=0.8$, $\mathcal{P}^1$

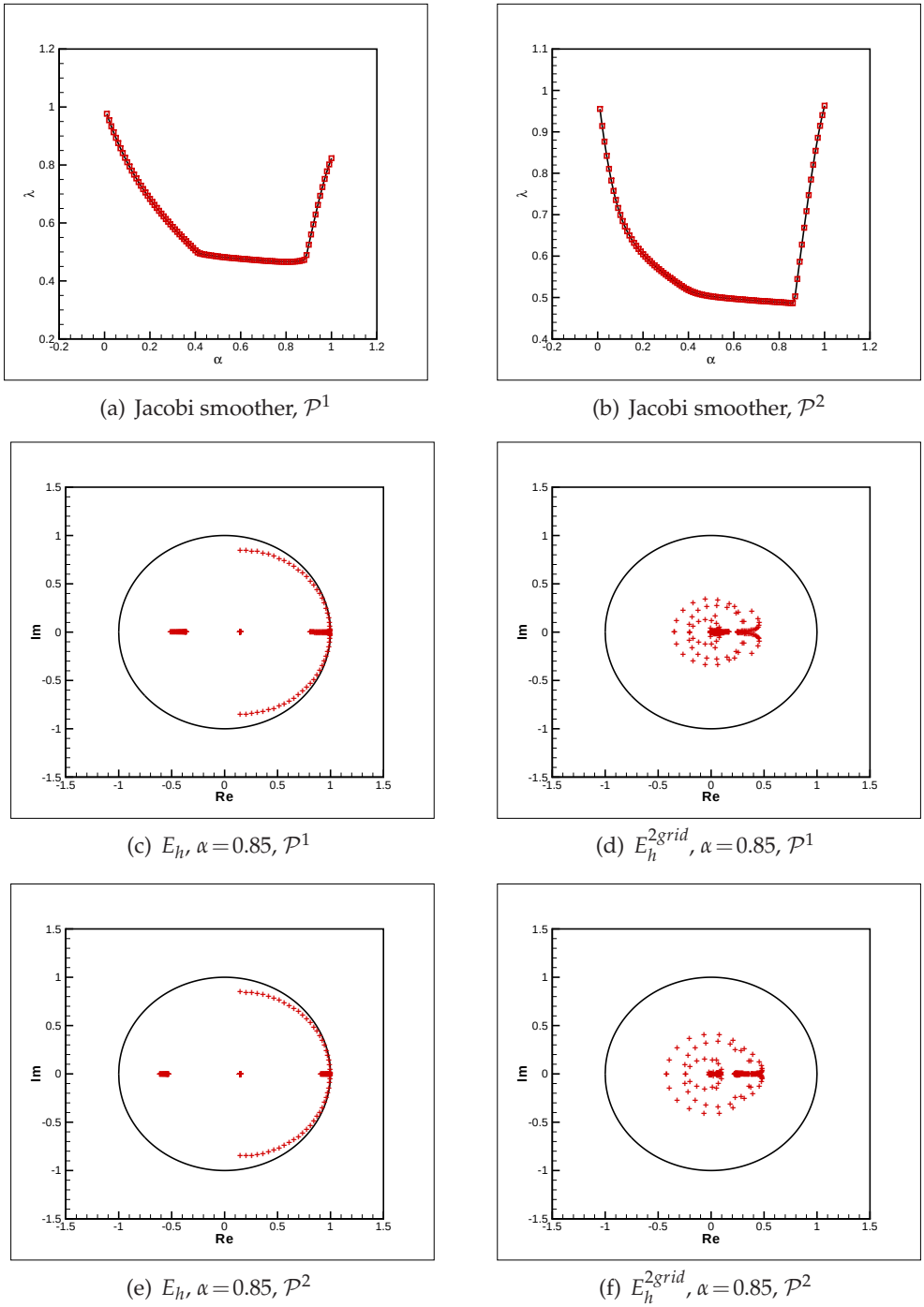


(e) E_h, $\alpha=0.6$, $\mathcal{P}^2$



(f) E_h^{2grid}, $\alpha=0.6$, $\mathcal{P}^2$

Figure 3: Method 3X, cell number $N=128$, $\Delta t=0.1\Delta x$, damped Gauss-Seidel smoother, $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation. Figs. 3(a)-3(b) show the asymptotic convergence factor λ changes with the damping parameter α. Figs. 3(c)-3(f) show the eigenvalue spectra of E_h and E_h^{2grid}.

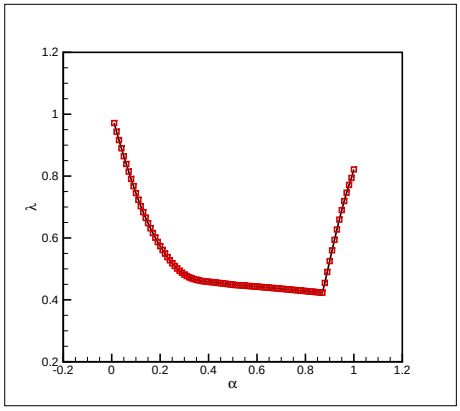


(a) Jacobi smoother, $\mathcal{P}^1$ (b) Jacobi smoother, $\mathcal{P}^2$

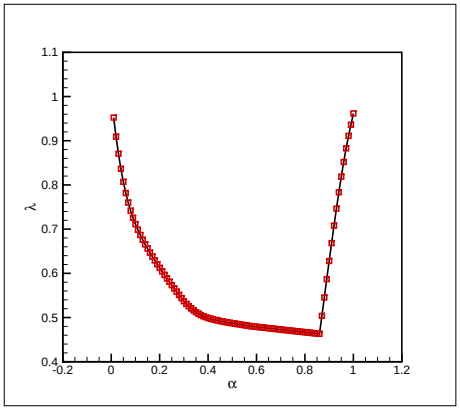
(c) E_h, $\alpha=0.85$, $\mathcal{P}^1$ (d) E_h^{2grid}, $\alpha=0.85$, $\mathcal{P}^1$

(e) E_h, $\alpha=0.85$, $\mathcal{P}^2$ (f) E_h^{2grid}, $\alpha=0.85$, $\mathcal{P}^2$

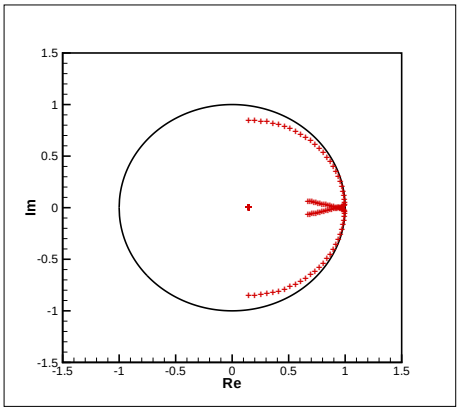
Figure 4: Method X-2X, cell number $N=128$, $\Delta t=0.1\Delta x$, damped Jacobi smoother, $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation. Figs. 4(a)-4(b) show the asymptotic convergence factor λ changes with the damping parameter α. Figs. 4(c)-4(f) show the eigenvalue spectra of E_h and E_h^{2grid}.



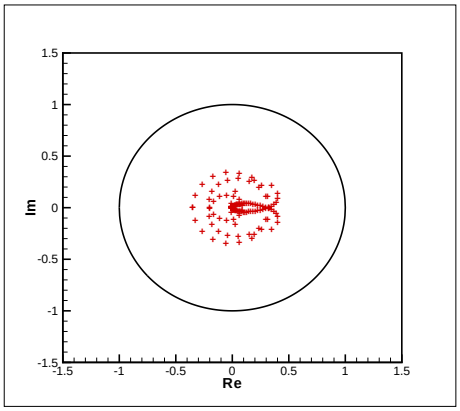
(a) Gauss-Seidel smoother, $\mathcal{P}^1$



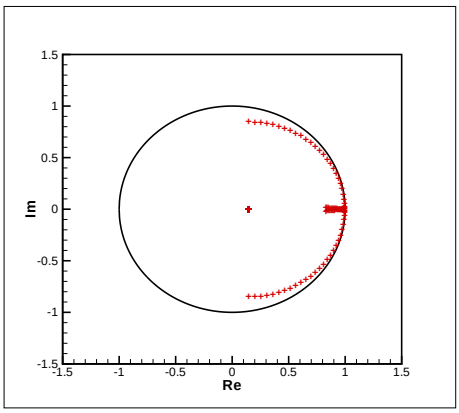
(b) Gauss-Seidel smoother, $\mathcal{P}^2$



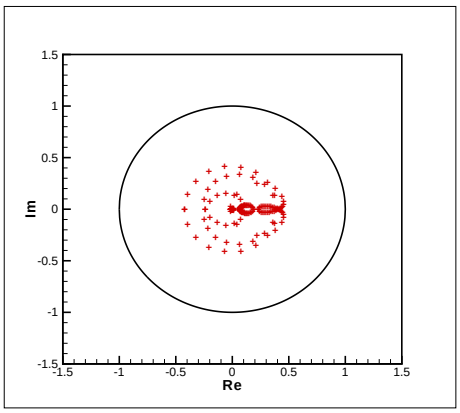
(c) E_h, $\alpha=0.85$, $\mathcal{P}^1$



(d) E_h^{2grid}, $\alpha=0.85$, $\mathcal{P}^1$



(e) E_h, $\alpha=0.85$, $\mathcal{P}^2$



(f) E_h^{2grid}, $\alpha=0.85$, $\mathcal{P}^2$

Figure 5: Method X-2X, cell number $N=128$, $\Delta t=0.1\Delta x$, damped Gauss-Seidel smoother, $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation. Figs. 5(a)-5(b) show the asymptotic convergence factor λ changes with the damping parameter α. Figs. 5(c)-5(f) show the eigenvalue spectra of E_h and E_h^{2grid}.

are integrated by an L-stable, stiffly-accurate, singly diagonally implicit Runge-Kutta method while the non-stiff terms are integrated with a traditional explicit Runge-Kutta method. For a detailed description of the methods as well as their implementation and applications, we refer the readers to [6].

We adopt the third order ARK method in this paper and from Eq. (3.14a), it is necessary to solve three system of linear equations at each time step. In order to solve these linear equations efficiently, we apply the MG method to solve them. Then by Eq. (3.14b), we can get $\boldsymbol{u}^{n+1}$.

4 The fifth-order KdV type equations

In this section, we apply the LDG spatial discretization to the fifth-order KdV type equation (1.2). To ease the presentation, we consider the simpler one-dimensional linear problems

$$u_t+u_{xxxxx}=0. \tag{4.1}$$

We refer the readers to [15,19] for high-dimensional and nonlinear problems. To define the LDG method to Eq. (4.1), we rewrite it as a first order system:

$$u_t+p_x=0,\quad p-q_x=0,\quad q-s_x=0,\quad s-r_x=0,\quad r-u_x=0. \tag{4.2}$$

To simplify the notation, we still use u, p, q, s and r to denote the numerical solution. Applying the DG method to the system (4.2), we have the scheme: Find $u,p,q,s,r\in V_h^k$, such that, for all test functions $\phi,\theta,\eta,\xi,\varphi\in V_h^k$, we have

$$(u_t,\phi)_K-(p,\phi_x)_K+(\hat{p},\phi)_{\partial K}=0, \tag{4.3a}$$
$$(p,\theta)_K+(q,\theta_x)_K-(\hat{q},\theta)_{\partial K}=0, \tag{4.3b}$$
$$(q,\eta)_K+(s,\eta_x)_K-(\hat{s},\eta)_{\partial K}=0, \tag{4.3c}$$
$$(s,\xi)_K+(r,\xi_x)_K-(\hat{r},\xi)_{\partial K}=0, \tag{4.3d}$$
$$(r,\varphi)_K+(u,\varphi_x)_K-(\hat{u},\varphi)_{\partial K}=0. \tag{4.3e}$$

Following the discussion in [15,19], we can take the simple choices for the numerical fluxes such that

$$\hat{p}|_e=p_L,\quad \hat{q}|_e=q_R,\quad \hat{s}|_e=s_L,\quad \hat{r}|_e=r_R,\quad \hat{u}|_e=u_L. \tag{4.4}$$

Following the same method as described in Section 3, we have three methods to obtain $\boldsymbol{u}^{n+1}$ from the known $\boldsymbol{u}^n$.

- **Method 5X**

  We choose a local basis in cell K, then $\boldsymbol{r}$, $\boldsymbol{s}$, $\boldsymbol{q}$ and $\boldsymbol{p}$ can be eliminated from Eqs. (4.3e),

(4.3d), (4.3c) and (4.3b), respectively, by simply inverting a small mass matrix in each case. Then we get an ODE

$$\boldsymbol{u}_t = L(\boldsymbol{u}). \tag{4.5}$$

The backward Euler time marching method is applied to Eq. (4.5) and we obtain a linear system

$$K\boldsymbol{u}^{n+1} = \boldsymbol{f}, \tag{4.6}$$

where $\boldsymbol{f}$ is the corresponding right hand side vector consisting of $\boldsymbol{u}^n$ and K can be obtained as in Section 3.

- **Method 3X-2X**

  In this method, we eliminate $\boldsymbol{q}$, $\boldsymbol{p}$ and $\boldsymbol{r}$ from Eqs. (4.3c), (4.3b) and (4.3e), respectively, then we get

$$\begin{cases} \boldsymbol{u}_t = L_1(\boldsymbol{s}), \\ \boldsymbol{s} = L_2(\boldsymbol{u}). \end{cases} \tag{4.7}$$

  We apply the backward Euler time marching method to Eq. (4.7) and obtain a system of two coupled equations for $[\boldsymbol{u}^{n+1}, \boldsymbol{s}^{n+1}]$

$$G\boldsymbol{U} = \boldsymbol{F}, \tag{4.8}$$

  where $\boldsymbol{U} = [\boldsymbol{u}^{n+1}, \boldsymbol{s}^{n+1}]^T$ and $\boldsymbol{F}$ is the corresponding right hand side vector consisting of $\boldsymbol{u}^n$ and $\boldsymbol{s}^n$.

- **Method X-2X-2X**

  We eliminate $\boldsymbol{q}$ and $\boldsymbol{r}$ from Eqs. (4.3c) and (4.3e), respectively. After the elimination of the auxiliary variables, we get

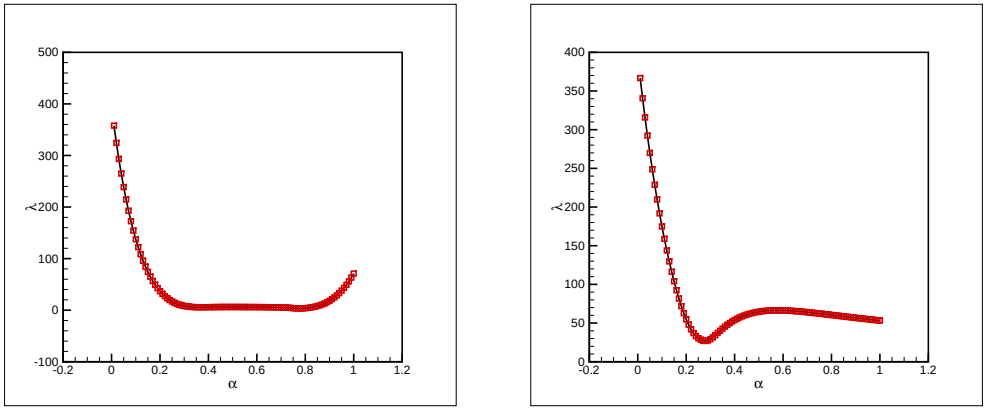
$$\begin{cases} \boldsymbol{u}_t = L_1(\boldsymbol{p}), \\ \boldsymbol{p} = L_2(\boldsymbol{s}), \\ \boldsymbol{s} = L_3(\boldsymbol{u}). \end{cases} \tag{4.9}$$

  The backward Euler time marching method is applied to Eq. (4.9) and we obtain a system of three coupled equations for $[\boldsymbol{u}^{n+1}, \boldsymbol{p}^{n+1}, \boldsymbol{s}^{n+1}]$

$$G\boldsymbol{U} = \boldsymbol{F}, \tag{4.10}$$

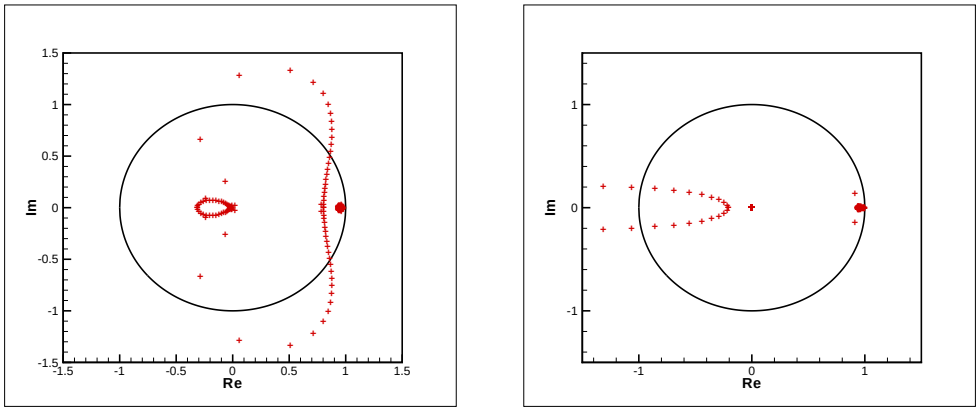
  where $\boldsymbol{U} = [\boldsymbol{u}^{n+1}, \boldsymbol{p}^{n+1}, \boldsymbol{s}^{n+1}]^T$ and $\boldsymbol{F}$ is the corresponding right hand side vector consisting of $\boldsymbol{u}^n$, $\boldsymbol{p}^n$ and $\boldsymbol{s}^n$.

We apply the MG method to solve the linear systems (4.6), (4.8) and (4.10). The convergence of the two-grid algorithm is analyzed by the local mode analysis. From Figs. 6-10, we have



(a) Jacobi smoother, $\mathcal{P}^1$

(b) Gauss-Seidel smoother, $\mathcal{P}^1$

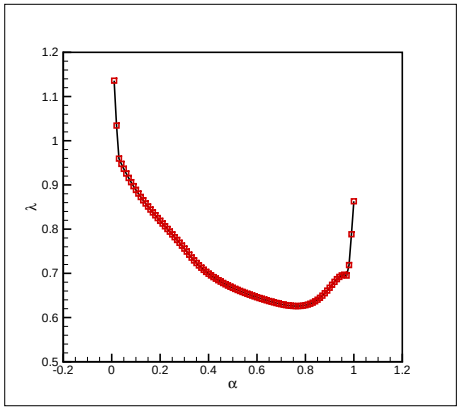


(c) E_h^{2grid}, $\alpha=0.8$, damped Jacobi smoother

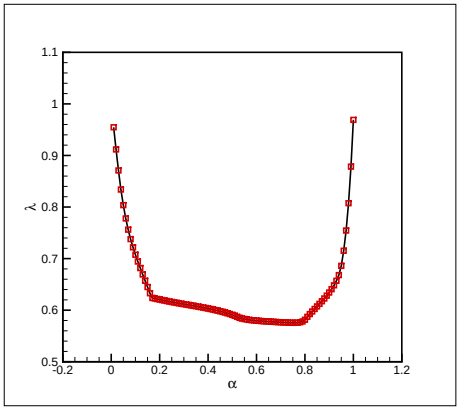
(d) E_h^{2grid}, $\alpha=0.8$, damped Gauss-Seidel smoother

Figure 6: Method 5X, cell number $N=128$, $\Delta t=0.1\Delta x$, damped Jacobi and Gauss-Seidel smoother, $\mathcal{P}^1$ approximation. Figs. 6(a)-6(b) show the asymptotic convergence factor λ changes with the damping parameter α. Figs. 6(c)-6(d) show the eigenvalue spectra of E_h^{2grid}.

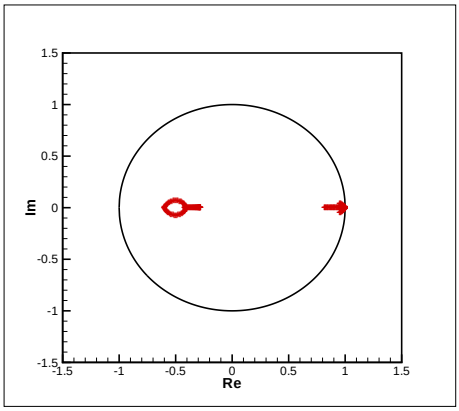
- The eigenvalue spectra of E_h^{2grid} for Method 5X is larger than 1 and the two-grid algorithm is not convergent. The reason is the high stiffness of the LDG spatial discretization operator and the condition number of the discretization matrix K is extremely large when we choose the large time step $\Delta t=0.1\Delta x$.
- The eigenvalue spectra of E_h is near 1 and the eigenvalue spectra of E_h^{2grid} is strictly less than 1, i.e. the two-grid algorithm is convergent for Method 3X-2X.
- Method X-2X-2X shows better convergence behavior than method 3X-2X with damped Gauss-Seidel smoother, while with damped Jacobi smoother, the convergence behavior for Method 3X-2X is better.



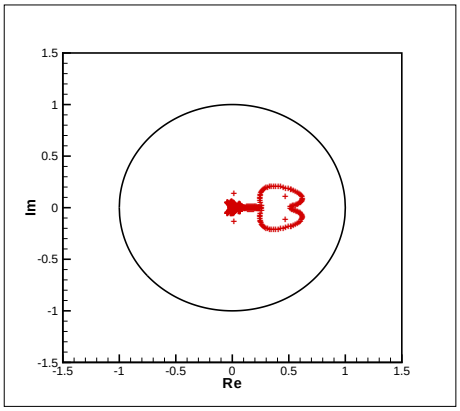
(a) Jacobi smoother, $\mathcal{P}^1$



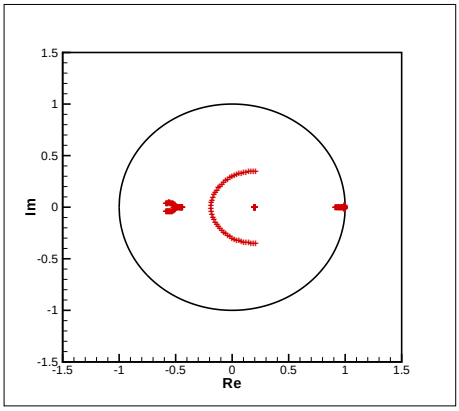
(b) Jacobi smoother, $\mathcal{P}^2$



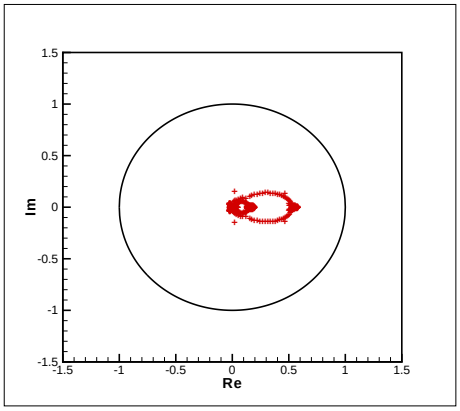
(c) E_h, $\alpha=0.8$, $\mathcal{P}^1$



(d) E_h^{2grid}, $\alpha=0.8$, $\mathcal{P}^1$

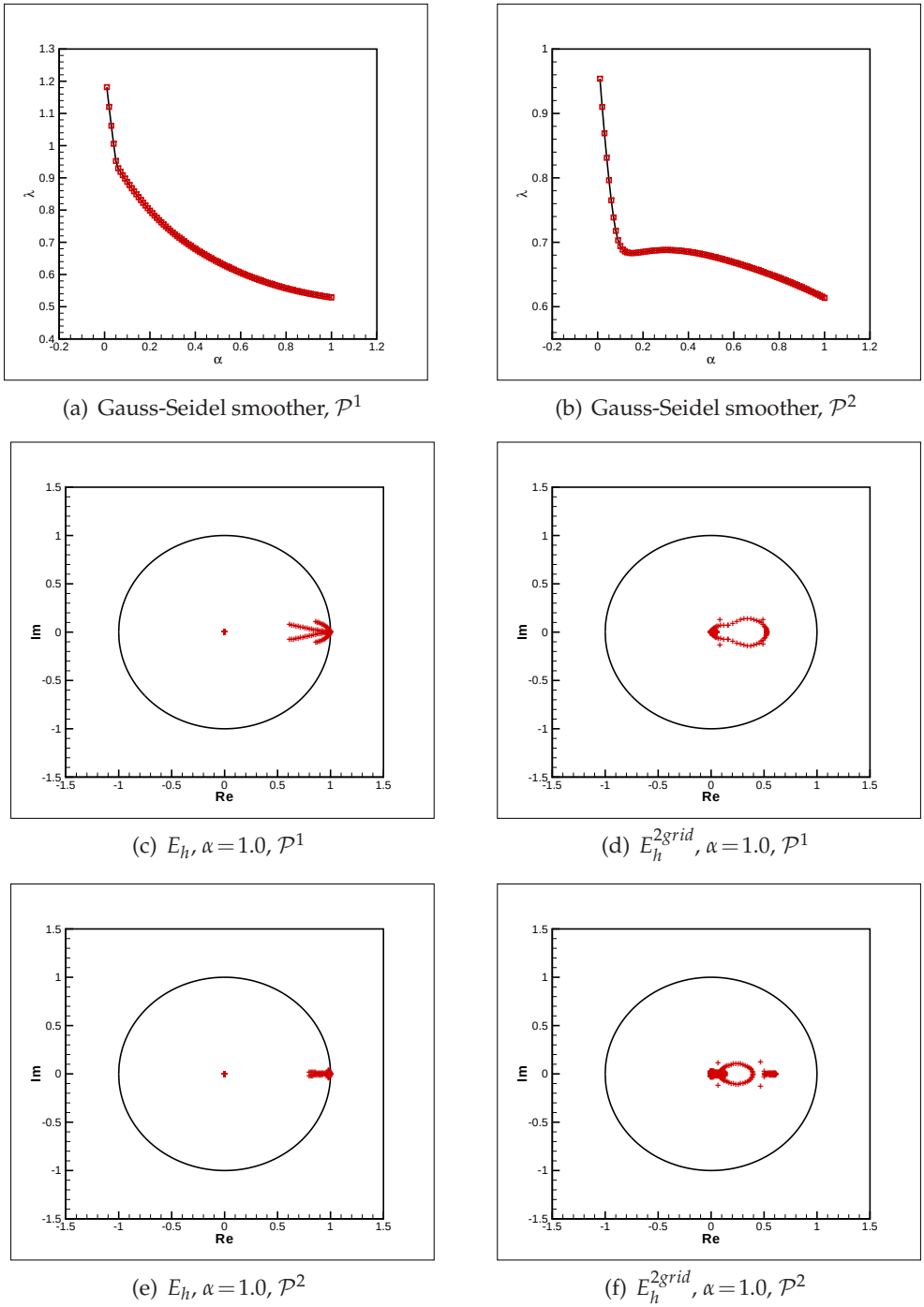


(e) E_h, $\alpha=0.8$, $\mathcal{P}^2$



(f) E_h^{2grid}, $\alpha=0.8$, $\mathcal{P}^2$

Figure 7: Method 3X-2X, cell number $N=128$, $\Delta t=0.1\Delta x$, damped Jacobi smoother, $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation. Figs. 7(a)-7(b) show the asymptotic convergence factor λ changes with the damping parameter α. Figs. 7(c)-7(f) show the eigenvalue spectra of E_h and E_h^{2grid}.



(a) Gauss-Seidel smoother, $\mathcal{P}^1$

(b) Gauss-Seidel smoother, $\mathcal{P}^2$

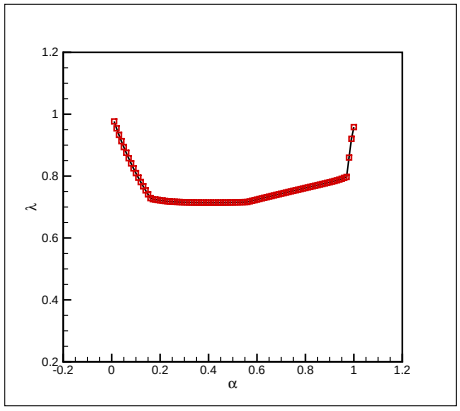
(c) E_h, $\alpha=1.0$, $\mathcal{P}^1$

(d) E_h^{2grid}, $\alpha=1.0$, $\mathcal{P}^1$

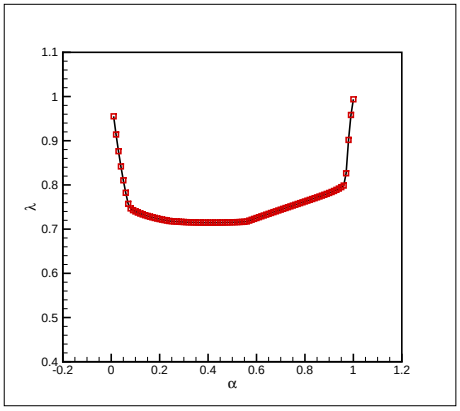
(e) E_h, $\alpha=1.0$, $\mathcal{P}^2$

(f) E_h^{2grid}, $\alpha=1.0$, $\mathcal{P}^2$

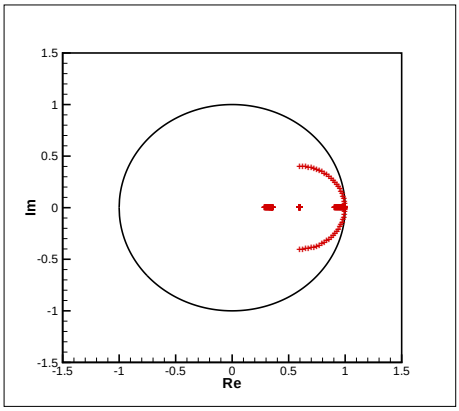
Figure 8: Method 3X-2X, cell number $N=128$, $\Delta t=0.1\Delta x$, damped Gauss-Seidel smoother, $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation. Figs. 8(a)-8(b) show the asymptotic convergence factor λ changes with the damping parameter α. Figs. 8(c)-8(f) show the eigenvalue spectra of E_h and E_h^{2grid}.



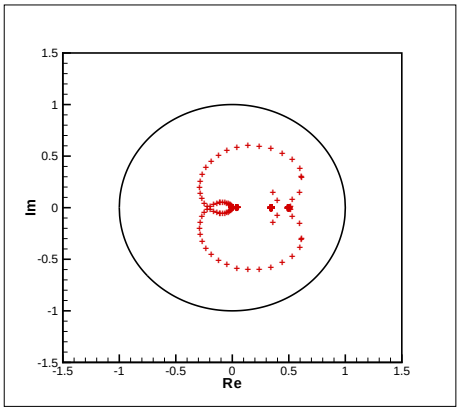
(a) Jacobi smoother, $\mathcal{P}^1$



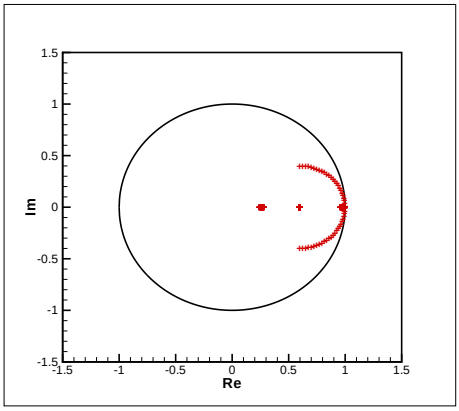
(b) Jacobi smoother, $\mathcal{P}^2$



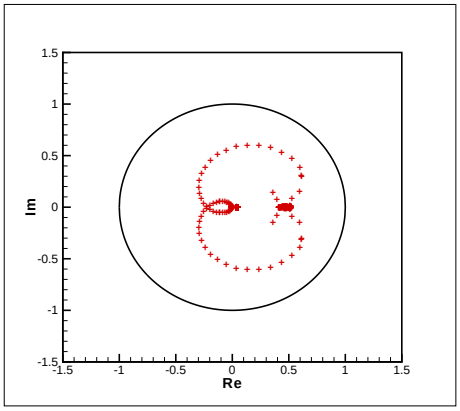
(c) E_h, $\alpha=0.4$, $\mathcal{P}^1$



(d) E_h^{2grid}, $\alpha=0.4$, $\mathcal{P}^1$

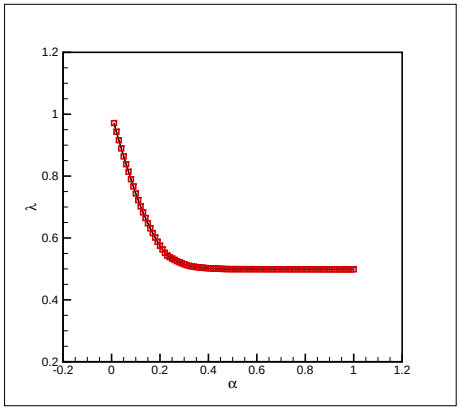


(e) E_h, $\alpha=0.4$, $\mathcal{P}^2$

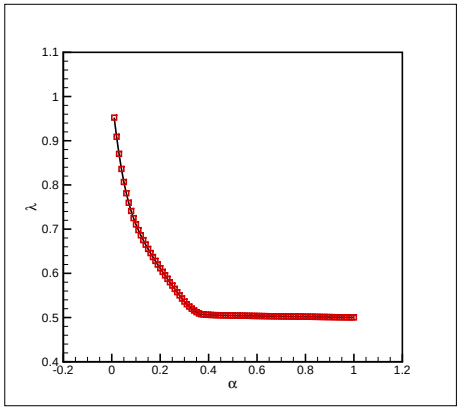


(f) E_h^{2grid}, $\alpha=0.4$, $\mathcal{P}^2$

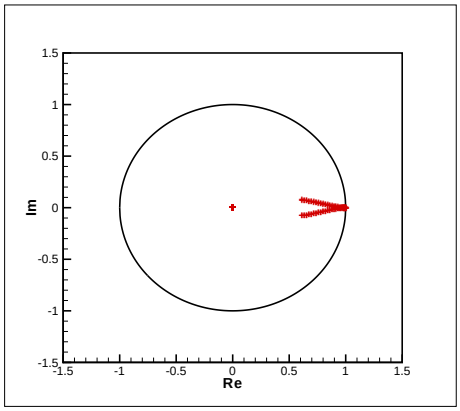
Figure 9: Method X-2X-2X, cell number $N=128$, $\Delta t=0.1\Delta x$, damped Jacobi smoother, $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation. Figs. 9(a)-9(b) show the asymptotic convergence factor λ changes with the damping parameter α. Figs. 9(c)-9(f) show the eigenvalue spectra of E_h and E_h^{2grid}.



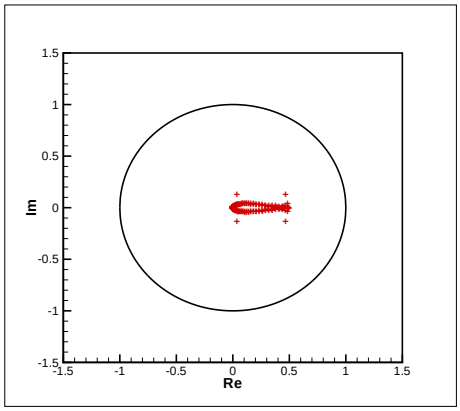
(a) Gauss-Seidel smoother, $\mathcal{P}^1$



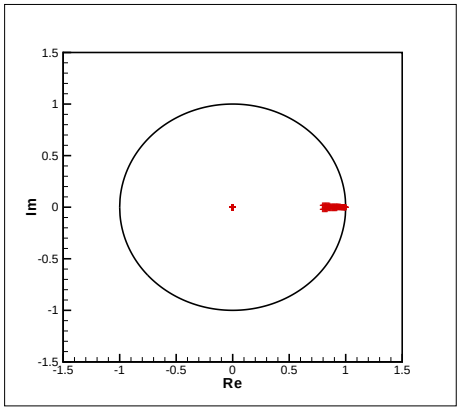
(b) Gauss-Seidel smoother, $\mathcal{P}^2$



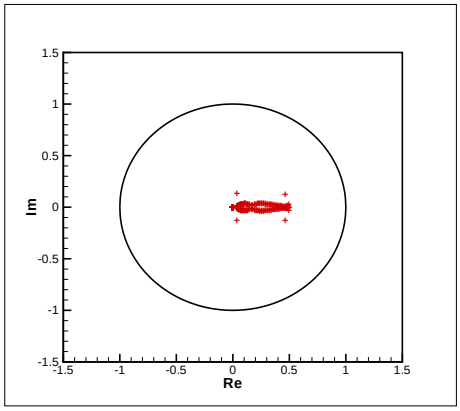
(c) E_h, $\alpha=1.0$, $\mathcal{P}^1$



(d) E_h^{2grid}, $\alpha=1.0$, $\mathcal{P}^1$



(e) E_h, $\alpha=1.0$, $\mathcal{P}^2$



(f) E_h^{2grid}, $\alpha=1.0$, $\mathcal{P}^2$

Figure 10: Method X-2X-2X, cell number $N=128$, $\Delta t=0.1\Delta x$, damped Gauss-Seidel smoother, $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation. Figs. 10(a)-10(b) show the asymptotic convergence factor λ changes with the damping parameter α. Figs. 10(c)-10(f) show the eigenvalue spectra of E_h and E_h^{2grid}.

5 The general odd-order linear PDEs

In this section, we consider the general odd-order linear PDEs

$$u_t+u_x^{(2m+1)}=0, \tag{5.1}$$

and give the generalization of the LDG scheme. We first rewrite (5.1) as a first order system:

$$u_t+(p_{2m})_x=0, \quad p_{2m}=(p_{2m-1})_x,\cdots,p_2=(p_1)_x, \quad p_1=u_x. \tag{5.2}$$

The LDG method to solve the system (5.2) is as follows: Find $u,p_{2m},\cdots,p_2,p_1\in V_h^k$, such that, for all test functions $\phi,\theta,\cdots,\eta,\xi\in V_h^k$, we have

$$\begin{aligned}
&(u_t,\phi)_K-(p_{2m},\phi_x)_K+(\hat{p}_{2m},\phi)_{\partial K}=0, && \text{(5.3a)}\\
&(p_{2m},\theta)_K+(p_{2m-1},\theta_x)_K-(\hat{p}_{2m-1},\theta)_{\partial K}=0, && \text{(5.3b)}\\
&\quad\cdots\\
&(p_2,\eta)_K+(p_1,\eta_x)_K-(\hat{p}_1,\eta)_{\partial K}=0, && \text{(5.3c)}\\
&(p_1,\xi)_K+(u,\xi_x)_K-(\hat{u},\xi)_{\partial K}=0. && \text{(5.3d)}
\end{aligned}$$

Following the same method as for the third-order KdV equation described in Section 3. We have the following two methods to obtain $\boldsymbol{u}^{n+1}$ from the known $\boldsymbol{u}^n$:

- **Method 3X-2X-2X$\cdots$-2X**

  We choose a local basis in cell K, then $\boldsymbol{p}_{2m}$, $\boldsymbol{p}_{2m-1}$, $\boldsymbol{p}_{2m-3}$,$\cdots$, $\boldsymbol{p}_1$ can be eliminated by simply inverting a small mass matrix in each case. Then we get

$$\begin{cases}
\boldsymbol{u}_t=L_1(\boldsymbol{p}_{2m-2}),\\
\boldsymbol{p}_{2m-2}=L_2(\boldsymbol{p}_{2m-4}),\\
\quad\cdots\\
\boldsymbol{p}_2=L_m(\boldsymbol{u}).
\end{cases} \tag{5.4}$$

  The backward Euler time marching method is applied to Eq. (5.4) and we obtain a system of m coupled equations for $[\boldsymbol{u}^{n+1},\boldsymbol{p}_{2m-2}^{n+1},\cdots,\boldsymbol{p}_2^{n+1}]$

$$G\boldsymbol{U}=\boldsymbol{F}, \tag{5.5}$$

  where $\boldsymbol{U}=[\boldsymbol{u}^{n+1},\boldsymbol{p}_{2m-2}^{n+1},\cdots,\boldsymbol{p}_2^{n+1}]^T$ and $\boldsymbol{F}$ is the corresponding right hand side vector consisting of $\boldsymbol{u}^n$, $\boldsymbol{p}_{2m-2}^n$,$\cdots$, $\boldsymbol{p}_2^n$.

- **Method X-2X-2X$\cdots$-2X**

In this method, we eliminate $\boldsymbol{p}_{2m-1}$, $\boldsymbol{p}_{2m-3}$,$\cdots$, $\boldsymbol{p}_1$, then we get

$$\begin{cases} \boldsymbol{u}_t = L_1(\boldsymbol{p}_{2m}), \\ \boldsymbol{p}_{2m} = L_2(\boldsymbol{p}_{2m-2}), \\ \quad \cdots \\ \boldsymbol{p}_2 = L_{m+1}(\boldsymbol{u}). \end{cases} \tag{5.6}$$

We apply the backward Euler time marching method to Eq. (5.6) and obtain a system of $m+1$ coupled equations for $[\boldsymbol{u}^{n+1},\boldsymbol{p}_{2m}^{n+1},\boldsymbol{p}_{2m-2}^{n+1},\cdots,\boldsymbol{p}_2^{n+1}]$

$$G\boldsymbol{U} = \boldsymbol{F}, \tag{5.7}$$

where $\boldsymbol{U} = [\boldsymbol{u}^{n+1},\boldsymbol{p}_{2m}^{n+1},\boldsymbol{p}_{2m-2}^{n+1},\cdots,\boldsymbol{p}_2^{n+1}]^T$ and $\boldsymbol{F}$ is the corresponding right hand side vector consisting of $\boldsymbol{u}^n$, $\boldsymbol{p}_{2m}^n$, $\boldsymbol{p}_{2m-2}^n$,$\cdots$, $\boldsymbol{p}_2^n$.

The MG method is applied to solve the linear systems (5.5) and (5.7). What we should notice is that the notation K, $\boldsymbol{f}$, L, L_i, G, $\boldsymbol{U}$ and $\boldsymbol{F}$ are not the same in each occurrence.

Remark 5.1. To obtain a high order accuracy in time for the fifth-order KdV type equations and general odd-order linear PDEs, we can adopt the ARK method and numerical experiments show that it is efficient. As to the resulting linear systems, we apply the MG method to solve them.

6 Numerical results

In this section, we perform numerical experiments of the LDG scheme coupled with the third order ARK method to linear and nonlinear KdV type PDEs in one, two and three space dimensions, then we extend these methods to the seventh-order linear PDE. The resulting linear systems are solved by the MG method. For spatial discretization we use uniform meshes. In our numerical experiments, we choose the numbers of pre- and post- relaxations are equal, and set to be $\nu_1 = \nu_2 = 3$. As for the damping parameter α for damped Jacobi and damped Gauss-Seidel smoother, we just choose the optimal value by the Figures which show the asymptotic convergence factor λ changes with the damping parameter α. The accuracy and order are listed in Tables to confirm that the ARK time integration method is high order in time and the multigrid solver maintains the accuracy and order.

6.1 One space dimension

We show the accuracy and order for the KdV type equations containing third- and fifth-order spatial derivative in Examples 6.1-6.4. To demonstrate the convergence and optimal or sub-optimal complexity of the MG solver, we plot figures on the reduction in the norm

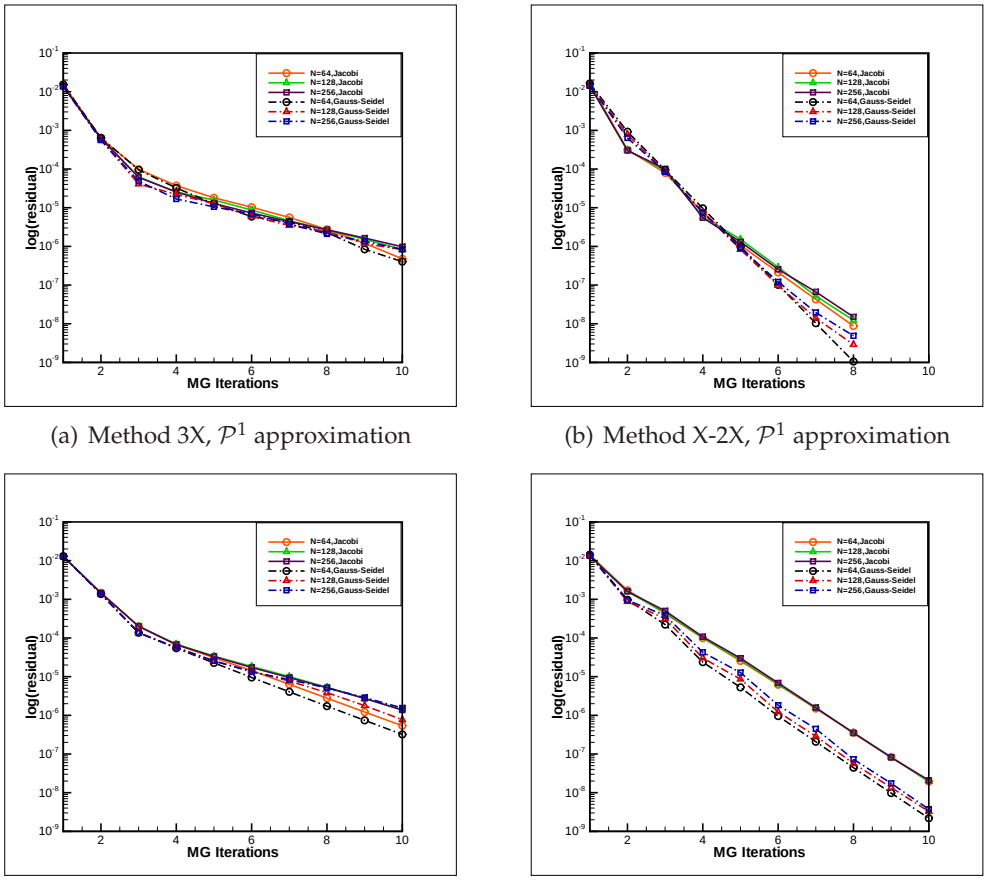
of the residual per V-cycle iteration with damped Jacobi smoother and damped Gauss-Seidel smoother, respectively. Then, we extend the MG solver to the seventh-order linear PDE in Example 6.5.

Example 6.1. We compute the linear KdV equation

$$u_t+u_{xxx}=0, \tag{6.1}$$

with an initial condition $u(x,0)=\sin(x)+\sin(2x)$ and a periodic boundary condition in $\Omega=(0,2\pi)$. The exact solution is given by $u(x,t)=\sin(x+t)+\sin(2x+8t)$. The L^2 and L^∞ errors, and the numerical order of accuracy at time $t=0.5$ are presented in Table 1. We can see that the method with $\mathcal{P}^k$ elements gives a $(k+1)$-th order of accuracy.

Fig. 11 shows the convergence of the MG solver and we can see optimal or sub-optimal complexity for Method X-2X and Method 3X has worse convergence behavior



(a) Method 3X, $\mathcal{P}^1$ approximation

(b) Method X-2X, $\mathcal{P}^1$ approximation

(c) Method 3X, $\mathcal{P}^2$ approximation

(d) Method X-2X, $\mathcal{P}^2$ approximation

Figure 11: Convergence rates for the MG solver for $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation for Example 6.1.

Table 1: Accuracy test for the equation $u_t+u_{xxx}=0$ at $t=0.5$, $\Delta t=0.1\Delta x$.

		Method 3X				Method X-2X			
	N	L^2 error	order	L^∞ error	order	L^2 error	order	L^∞ error	order
$\mathcal{P}^1$	16	3.41E-02	–	6.59E-02	–	3.41E-02	–	6.60E-02	–
	32	7.31E-03	2.22	2.08E-02	1.66	7.31E-03	2.22	2.08E-02	1.66
	64	1.74E-03	2.07	5.77E-03	1.85	1.74E-03	2.07	5.77E-03	1.85
	128	4.29E-04	2.01	1.50E-03	1.94	4.29E-04	2.01	1.50E-03	1.94
$\mathcal{P}^2$	16	2.78E-03	–	5.94E-03	–	2.78E-03	–	5.94E-03	–
	32	3.50E-04	2.99	7.90E-04	2.91	3.50E-04	2.99	7.90E-04	2.91
	64	4.41E-05	2.99	1.00E-04	2.98	4.41E-05	2.99	1.00E-04	2.98
	128	5.51E-06	3.00	1.26E-05	2.99	5.51E-06	3.00	1.26E-05	2.99

Table 2: The percentage of CPU time for third order ARK method and explicit RK method.

N	16	32	64	128
CPU time (ARK)/CPU time (RK)	22.18%	6.47%	1.92%	0.51%

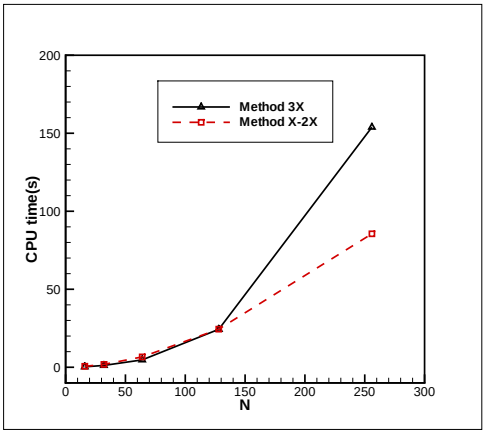


Figure 12: The CPU time changes with the cell number N of Method 3X and Method X-2X with $\mathcal{P}^2$ approximation for Example 6.1.

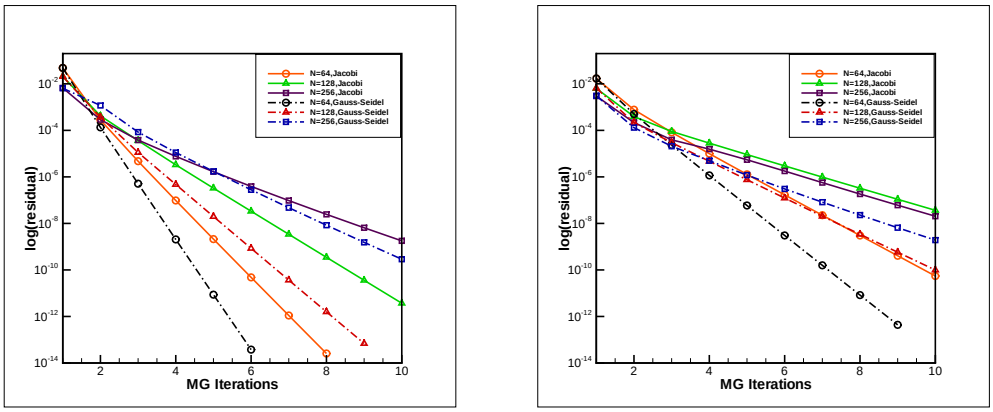
than Method X-2X. Thus, Method X-2X costs less CPU time, which agrees with the results in Fig. 12. What we should notice is that Method X-2X will require more memory for high-dimensional problems. Table 2 shows the percentage of CPU time for third order ARK method and explicit Runge-Kutta (RK) method and we have a conclusion that the ARK method is more efficient than explicit methods.

Example 6.2. We show an accuracy test for the KdV equation

$$u_t-3(u^2)_x+u_{xxx}=0, \tag{6.2}$$

Table 3: Accuracy test for the KdV equation $u_t-3(u^2)_x+u_{xxx}=0$ at $t=0.5$ with Method 3X.

	N	L^2 error	order	L^∞ error	order
$\mathcal{P}^1$	64	2.86E-01	–	5.71E-01	–
	128	8.99E-02	1.67	1.99E-01	1.52
	256	2.52E-02	1.83	6.06E-02	1.72
	512	6.64E-03	1.92	1.61E-02	1.91
$\mathcal{P}^2$	64	1.37E-01	–	3.21E-01	–
	128	2.07E-02	2.73	5.09E-02	2.66
	256	2.66E-03	2.96	6.67E-03	2.93
	512	3.35E-04	2.99	8.40E-04	2.99



(a) $\mathcal{P}^1$ approximation (b) $\mathcal{P}^2$ approximation

Figure 13: Convergence rates for the MG solver for $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation for Example 6.2 with Method 3X.

with the initial condition $u(x,0)=-2\text{sech}^2(x)$. The exact solution is given by

$$u(x,t)=-2\text{sech}^2(x-4t). \tag{6.3}$$

The computational domain is given by $[-20,22]$, and we use a periodic boundary condition. The time step is taken as $\Delta t=C\frac{\Delta x}{\max|6u(x,0)|}$, where C is the CFL number of the convection term associated with the stability of DG discretization. As to time discretization, we apply the third order implicit-explicit ARK methods and the resulting systems are solved by the MG method. We can see in Table 3 that the method with $\mathcal{P}^k$ elements gives $(k+1)$-th order of accuracy. To illustrate the sub-optimal complexity of the MG solver, we present the convergence rates of the method in Fig. 13.

Example 6.3. We consider the linear equation

$$u_t+u_{xxxxx}=0, \tag{6.4}$$

Table 4: Accuracy test for the equation $u_t+u_{xxxxx}=0$ at $t=0.5$, $\Delta t=0.1\Delta x$.

		Method 3X-2X				Method X-2X-2X			
	N	L^2 error	order	L^∞ error	order	L^2 error	order	L^∞ error	order
$\mathcal{P}^1$	16	3.53E-01	–	4.99E-01	–	3.53E-01	–	4.99E-01	–
	32	7.02E-02	2.32	1.07E-01	2.21	7.02E-02	2.32	1.07E-01	2.21
	64	1.00E-02	2.80	1.63E-02	2.71	1.00E-02	2.80	1.63E-02	2.71
	128	1.34E-03	2.90	2.38E-03	2.77	1.34E-03	2.90	2.38E-03	2.77
$\mathcal{P}^2$	16	3.19E-01	–	4.54E-01	–	3.19E-01	–	4.54E-01	–
	32	6.18E-02	2.36	8.77E-02	2.37	6.18E-02	2.36	8.77E-02	2.37
	64	8.73E-03	2.82	1.24E-02	2.82	8.73E-03	2.82	1.24E-02	2.82
	128	1.12E-03	2.96	1.59E-03	2.96	1.12E-03	2.96	1.59E-03	2.96

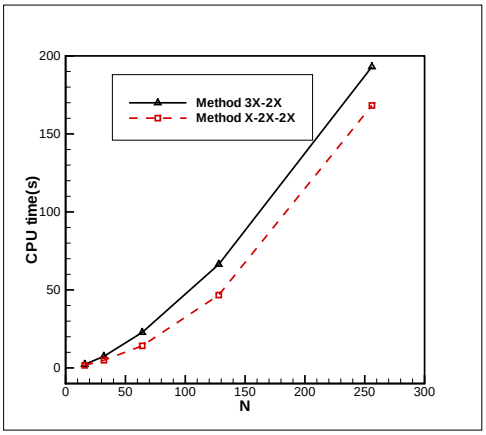
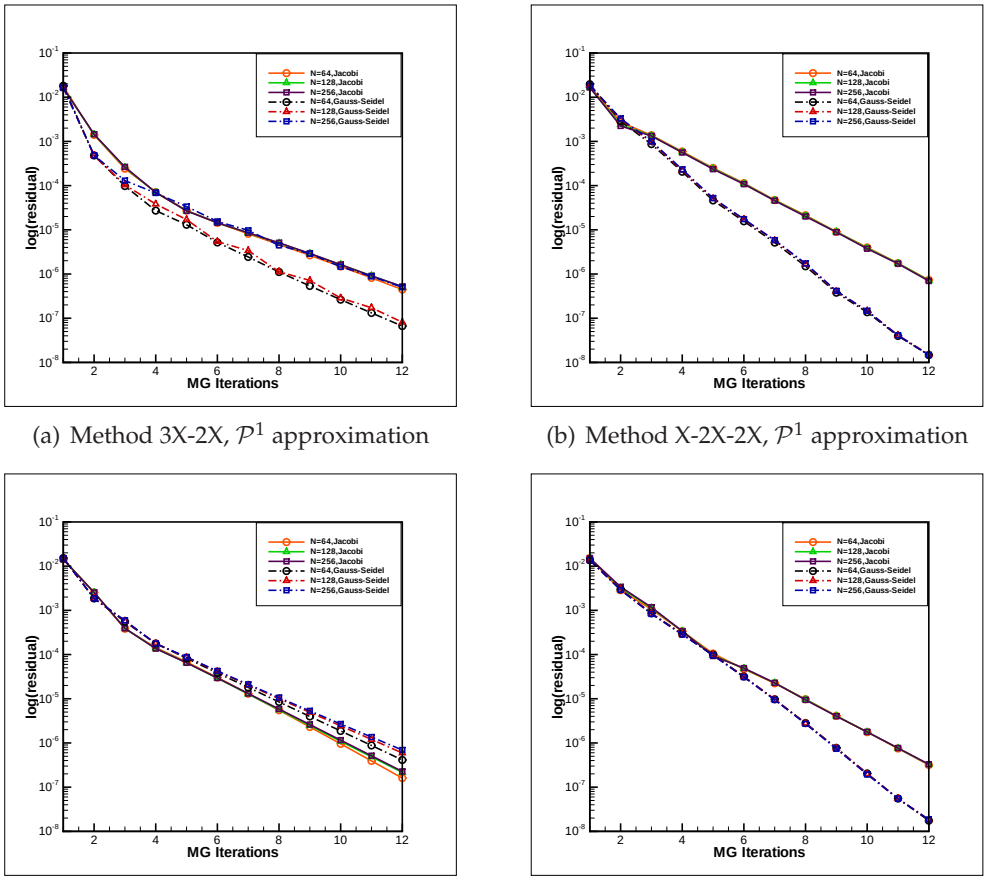


Figure 14: The CPU time changes with the cell number N of Method 3X-2X and Method X-2X-2X with $\mathcal{P}^2$ approximation for Example 6.3.

with an initial condition $u(x,0)=\sin(x)+\sin(2x)$ and a periodic boundary condition in $\Omega=(0,2\pi)$. The exact solution is given by $u(x,t)=\sin(x-t)+\sin(2x-32t)$. The numerical errors and orders of accuracy at time $t=0.5$ can be found in Table 4. We can see that the method with $\mathcal{P}^k$ elements gives a $(k+1)$-th order of accuracy. Method 5X is not convergent, thus we consider Method 3X-2X and Method X-2X-2X. The convergence behavior of the MG solver is shown in Fig. 15, which suggests that Method X-2X-2X converges faster than Method 3X-2X. Thus, Method X-2X-2X costs less CPU time, which agrees with the results in Fig. 14. What we should notice is that Method X-2X-2X will require more memory for high-dimensional problems.

Example 6.4. We show an accuracy test for the Kawahara equation

$$u_t+uu_x+u_{xxx}-u_{xxxxx}=0, \tag{6.5}$$



(a) Method 3X-2X, $\mathcal{P}^1$ approximation

(b) Method X-2X-2X, $\mathcal{P}^1$ approximation

(c) Method 3X-2X, $\mathcal{P}^2$ approximation

(d) Method X-2X-2X, $\mathcal{P}^2$ approximation

Figure 15: Convergence rates for the MG solver for $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation for Example 6.3.

with the exact solution

$$u(x,t)=\frac{105}{169}\text{sech}^4\left[\frac{1}{2\sqrt{13}}\left(x-\frac{36t}{169}\right)\right]. \tag{6.6}$$

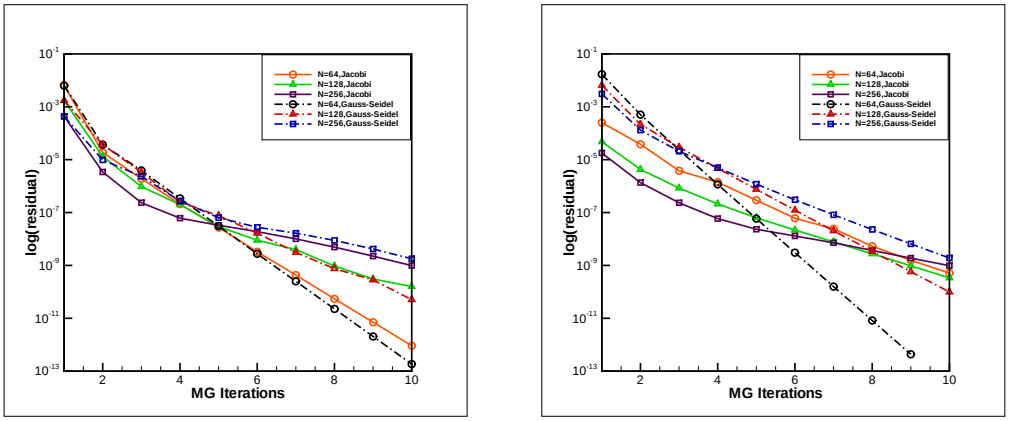
The computational domain is given by $[-35,35]$, and we use a periodic boundary condition. The time step is taken as $\Delta t = C\frac{\Delta x}{\max|u(x,0)|}$, where C is the CFL number of the convection term associated with the stability of DG discretization. We can see in Table 5 that the method with $\mathcal{P}^k$ elements gives $(k+1)$-th order of accuracy. The convergence rates of the MG solver for $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation are shown in Fig. 16, which suggests that the solver is of sub-optimal complexity.

Example 6.5. We consider the linear seventh-order equation

$$u_t+u_{xxxxxxx}=0, \tag{6.7}$$

Table 5: Accuracy test for the Kawahara equation $u_t+uu_x+u_{xxx}-u_{xxxxx}=0$ at $t=1.0$ with Method 3X-2X.

	N	L^2 error	order	L^∞ error	order
$\mathcal{P}^1$	16	4.26E-01	–	3.49E-01	–
	32	1.30E-01	1.71	1.20E-01	1.54
	64	3.43E-02	1.92	3.37E-02	1.83
	128	8.69E-03	1.98	8.82E-03	1.93
$\mathcal{P}^2$	16	2.52E-01	–	2.30E-01	–
	32	3.29E-02	2.94	3.19E-02	2.85
	64	4.24E-03	2.96	4.42E-03	2.85
	128	5.35E-04	2.99	5.73E-04	2.95



(a) $\mathcal{P}^1$ approximation (b) $\mathcal{P}^2$ approximation

Figure 16: Convergence rates for the MG solver for $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation for Example 6.4 with Method 3X-2X.

with an initial condition $u(x,0)=\sin(x)$ and a periodic boundary condition in $\Omega=(0,2\pi)$. The exact solution is given by $u(x,t)=\sin(x+t)$. The L^2 and L^∞ errors, and the numerical order of accuracy at time $t=0.5$ are presented in Table 6. We can see that the method with $\mathcal{P}^k$ elements gives a $(k+1)$-th order of accuracy. The errors are comparable between Method 3X-2X-2X and Method X-2X-2X-2X.

6.2 Two space dimension

We show the same information for two-dimensional problems as for one-dimensional PDEs in Examples 6.6-6.8, to conclude that the MG solver is also efficient for two-dimensional linear and nonlinear PDEs. But the MG solver shows poor convergence behavior, the reason is that the condition number of the discretization matrix is extremely large because of the high stiffness of the LDG spatial discretization operator for high-dimensional problems.

Table 6: Accuracy test for the equation $u_t+u_{xxxxxxx}=0$ at $t=0.5$, $\Delta t=0.1\Delta x$.

		Method 3X-2X-2X				Method X-2X-2X-2X			
	N	L^2 error	order	L^∞ error	order	L^2 error	order	L^∞ error	order
$\mathcal{P}^1$	16	3.18E-02	–	4.41E-02	–	3.18E-02	–	4.41E-02	–
	32	8.02E-03	1.99	1.11E-02	1.99	8.02E-03	1.99	1.11E-02	1.99
	64	2.01E-03	2.00	2.80E-03	1.99	2.01E-03	2.00	2.80E-03	1.99
	128	5.03E-04	2.00	7.01E-04	1.98	5.03E-04	2.00	7.01E-04	1.98
$\mathcal{P}^2$	16	3.64E-03	–	5.77E-03	–	3.64E-03	–	5.77E-03	–
	32	4.57E-04	2.99	7.22E-04	3.00	4.57E-04	2.99	7.22E-04	3.00
	64	5.72E-05	3.00	9.06E-05	2.99	5.72E-05	3.00	9.06E-05	2.99
	128	7.18E-06	2.99	1.22E-05	2.89	7.15E-06	3.00	1.13E-05	3.00

Example 6.6. We consider the linear KdV equation

$$u_t+u_{xxx}+u_{yyy}=0, \tag{6.8}$$

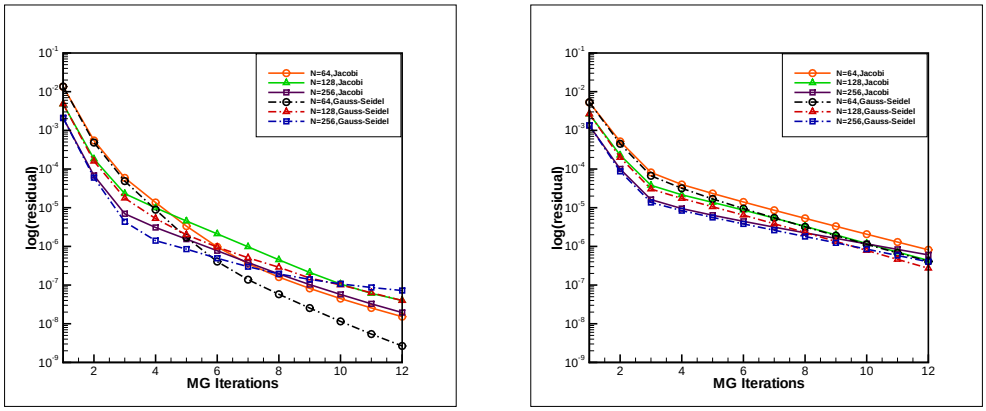
with an initial condition $u(x,y,0)=\sin(x+y)+\sin(2x+2y)$ and periodic boundary conditions in $\Omega=(0,2\pi)\times(0,2\pi)$. The exact solution is given by $u(x,y,t)=\sin(x+y+2t)+\sin(2x+2y+16t)$. The L^2 and L^∞ errors, and the numerical order of accuracy at time t=0.5 are presented in Table 7. We can see that the method with $\mathcal{P}^k$ elements gives a $(k+1)$-th order of accuracy. The reduction in the norm of the residual per V-cycle iteration is shown in Fig. 17, which suggests that the MG solver has sub-optimal complexity. Table 8 shows the percentage of CPU time for third order ARK method and explicit Runge-Kutta (RK) method and we can see that the ARK method is more efficient than explicit methods.

Table 7: Accuracy test for the equation $u_t+u_{xxx}+u_{yyy}=0$ at $t=0.5$ with Method 3X, $\Delta t=0.1\Delta x$.

	N	L^2 error	order	L^∞ error	order
$\mathcal{P}^1$	8	5.57E-01	–	8.55E-01	–
	16	1.56E-01	1.83	2.62E-01	1.70
	32	2.61E-02	2.58	5.31E-02	2.30
	64	4.44E-03	2.55	1.78E-02	1.57
$\mathcal{P}^2$	8	2.27E-01	–	3.81E-01	–
	16	3.46E-02	2.71	6.40E-02	2.57
	32	4.53E-03	2.94	9.16E-03	2.81
	64	5.76E-04	2.97	1.20E-03	2.93

Table 8: The percentage of CPU time for third order ARK method and explicit RK method.

N	8	16	32	64
CPU time (ARK)/CPU time (RK)	43.38%	15.61%	9.45%	4.01%



(a) $\mathcal{P}^1$ approximation (b) $\mathcal{P}^2$ approximation

Figure 17: Convergence rates of MG solver for $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation for Example 6.6 with Method 3X.

Example 6.7. We show an accuracy test for the Zakharov-Kuznetsov (ZK) equation. The steady progressive wave solutions of the ZK equation

$$u_t+uu_x+\varepsilon(u_{xxx}+u_{yyx})=0, \tag{6.9}$$

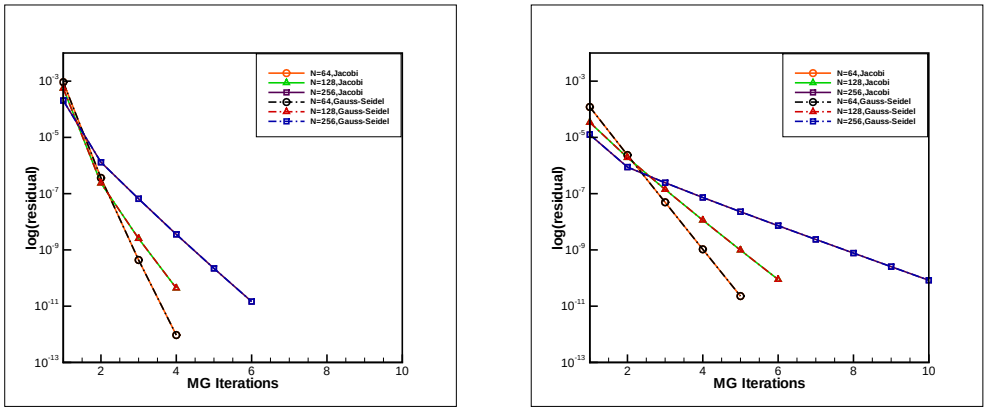
is of the form

$$u(x,y,t)=3c\text{sech}^2\left(0.5\sqrt{\frac{c}{\varepsilon}}((x-ct)\cos\theta+y\sin\theta)\right), \tag{6.10}$$

where θ is an inclined angle with respect to the x-axis, $c=0.01$, $\varepsilon=0.01$, $\theta=0$. The computational domain is given by $[-25,25]\times[-25,25]$, and we use periodic boundary conditions. We can see in Table 9 that the method with $\mathcal{P}^k$ elements gives $(k+1)$-th order of accuracy. The convergence rates of the MG solver is shown in Fig. 18, which suggests the sub-optimal complexity.

Table 9: Accuracy test for the equation $u_t+uu_x+0.01(u_{xxx}+u_{yyx})=0$ at $t=1.0$ with Method 3X, $\Delta t=0.1\Delta x$.

	N	L^2 error	order	L^∞ error	order
$\mathcal{P}^1$	32	1.84E-03	–	1.69E-03	–
	64	7.62E-04	1.28	1.08E-03	0.65
	128	1.92E-04	1.98	3.01E-04	1.85
	256	4.82E-05	1.99	7.73E-05	1.96
$\mathcal{P}^2$	32	4.42E-04	–	4.99E-04	–
	64	5.87E-05	2.91	6.81E-05	2.87
	128	7.08E-06	3.05	9.08E-06	2.91
	256	8.92E-07	2.99	1.16E-06	2.96



(a) $\mathcal{P}^1$ approximation (b) $\mathcal{P}^2$ approximation

Figure 18: Convergence rates of MG solver for $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation for Example 6.7 with Method 3X.

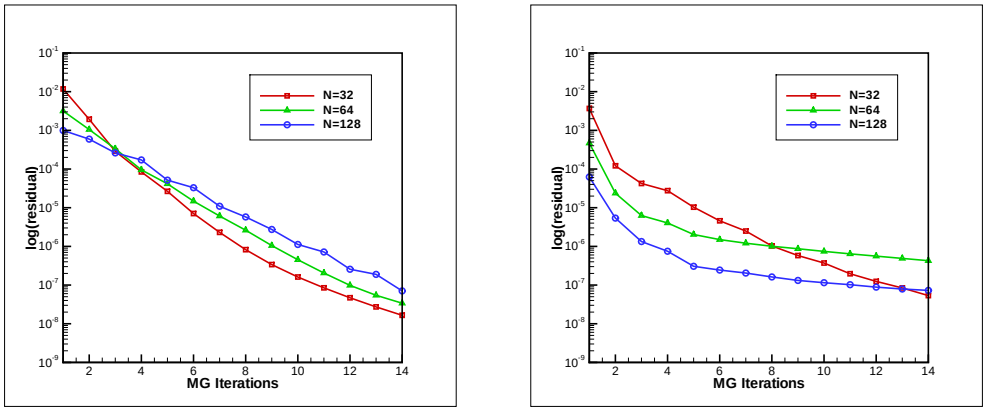
Example 6.8. We consider the linear equation

$$u_t+u_{xxxxx}+u_{yyyyy}=0, \tag{6.11}$$

with an initial condition $u(x,y,0)=\sin(x+y)$ and periodic boundary conditions in $\Omega=(0,2\pi)\times(0,2\pi)$. The exact solution is given by $u(x,y,t)=\sin(x+y-2t)$. The L^2 and L^∞ errors, and the numerical order of accuracy at time $t=0.5$ are presented in Table 10. We can see that the method with $\mathcal{P}^k$ elements gives a $(k+1)$-th order of accuracy. From Fig. 19, we can see the optimal or sub-optimal complexity of the MG solver for $\mathcal{P}^1$ approximation. For $\mathcal{P}^2$ approximation, the convergence of the method is worse, the reason is that the condition number of the discretization matrix is extremely large because of the high stiffness of the LDG spatial discretization operator.

Table 10: Accuracy test for the equation $u_t+u_{xxxxx}+u_{yyyyy}=0$ at $t=0.5$ with Method 3X-2X, $\Delta t=0.03\Delta x$.

	N	L^2 error	order	L^∞ error	order
$\mathcal{P}^1$	8	8.72E-02	–	1.23E-01	–
	16	2.47E-02	1.82	3.48E-02	1.82
	32	6.50E-03	1.93	9.19E-03	1.92
	64	1.66E-03	1.97	2.36E-03	1.96
$\mathcal{P}^2$	8	1.32E-02	–	1.84E-02	–
	16	1.75E-03	2.92	2.47E-03	2.90
	32	2.22E-04	2.98	3.16E-04	2.97
	64	2.68E-05	3.05	3.95E-05	3.00



(a) $\mathcal{P}^1$ approximation (b) $\mathcal{P}^2$ approximation

Figure 19: Convergence rates of MG solver with damped Gauss-Seidel smoother for $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation for Example 6.8 with Method 3X-2X.

6.3 Three space dimension

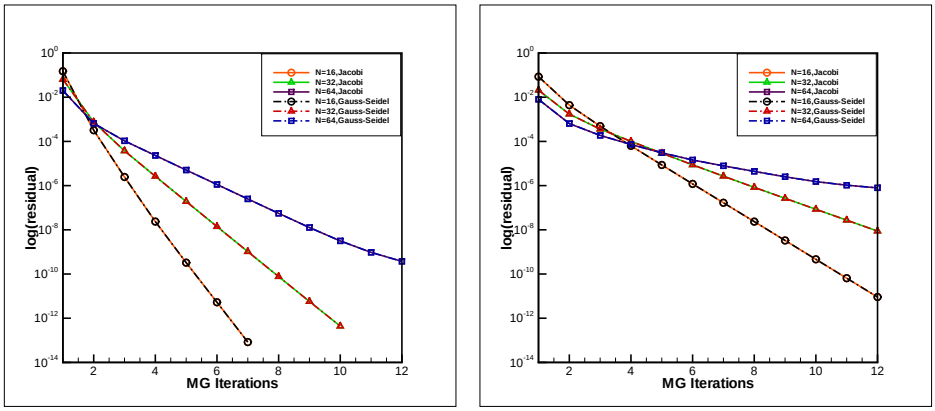
Example 6.9. We consider the linear KdV equation

$$u_t+u_{xxx}+u_{yyy}+u_{zzz}=0, \tag{6.12}$$

with an initial condition $u(x,y,z,0)=\sin(x+y+z)+\sin(2x+2y+2z)$ and periodic boundary conditions in $\Omega=(0,2\pi)\times(0,2\pi)\times(0,2\pi)$. The exact solution is given by $u(x,y,z,t)=\sin(x+y+z+3t)+\sin(2x+2y+2z+24t)$. The L^2 and L^∞ errors, and the numerical order of accuracy at time $t=0.5$ are presented in Table 11. We can see that the method with $\mathcal{P}^k$ elements gives a $(k+1)$-th order of accuracy. The convergence rates of the MG solver is shown in Fig. 20, which shows sub-optimal complexity for $\mathcal{P}^1$ approximation. The convergence behavior for $\mathcal{P}^2$ approximation is poor because of the high stiffness of the LDG spatial discretization operator for high-dimensional problems.

Table 11: Accuracy test for the equation $u_t+u_{xxx}+u_{yyy}+u_{zzz}=0$ at $t=0.5$ with Method 3X, $\Delta t=0.03\Delta x$.

	N	L^2 error	order	L^∞ error	order
$\mathcal{P}^1$	8	9.81E-01	–	1.68E+00	–
	16	3.81E-01	1.36	6.79E-01	1.31
	32	9.60E-02	1.98	1.69E-01	2.00
	64	2.89E-02	1.73	4.94E-02	1.77
$\mathcal{P}^2$	8	6.00E-01	–	1.02E+00	–
	16	1.51E-01	1.99	2.43E-01	2.07
	32	2.25E-02	2.75	3.54E-02	2.78
	64	2.92E-03	2.95	4.61E-03	2.94



(a) $\mathcal{P}^1$ approximation (b) $\mathcal{P}^2$ approximation

Figure 20: Convergence rates of MG solver for $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation for Example 6.9 with Method 3X.

7 Concluding remarks

In this paper, we have applied the MG solver to the system of algebraic equations arising from the LDG spatial discretization and ARK time marching method for the KdV type equations containing third- and fifth-order derivatives. The MG method and the ARK time marching method are also efficient for the general odd-order linear PDEs. Numerical examples are performed to verify that the ARK method is efficient in discretizing the LDG schemes in time, allowing time steps $\Delta t = \mathcal{O}(\Delta x)$ rather than the much more restrictive $\Delta t = \mathcal{O}(\Delta x^k)$ of explicit time discretizations for k-th order PDEs. The convergence of the MG algorithm is studied numerically by the local mode analysis. Numerical experiments for one-dimensional, two-dimensional and three-dimensional cases are given to illustrate the accuracy and capability of the LDG method coupled with the multigrid method.

The optimal or sub-optimal complexity of the MG solver for $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation are numerically shown. For $\mathcal{P}^2$ approximation, we see the convergence of the MG solver is not always good. The main possible reason is that the condition number of the discretization matrix is extremely large for high order spatial discretization. Therefore, it is necessary to introduce a preconditioner such that the discretization matrix is well conditioned. We leave this topic to our future work. Also, in this paper, we mainly focus on the numerical investigation, theoretical analysis is left to our future work.

Acknowledgments

Research of Y. Xu is supported by NSFC grant No. 11371342, No. 11031007, Fok Ying Tung Education Foundation No. 131003.

References

[1] F. Bassi, A. Ghidoni, S. Rebay and P. Tesini, High-order accurate p-multigrid discontinuous Galerkin solution of the Euler equations, International journal for numerical methods in fluids, 60 (2008), 847-865.
[2] F. Bassi, A. Ghidoni and S. Rebay, Optimal Runge-Kutta smoothers for the p-multigrid discontinuous Galerkin solution of the 1D Euler equations, J. Comput. Phys., 230 (2011), 4153-4175.
[3] J.L. Bona, H. Chen, O. Karakashian and Y. Xing, Conservative, discontinuous-Galerkin methods for the generalized Korteweg-de Vries equation, Mathematics of Computation, 82 (2013), 1401-1432.
[4] B. Cockburn and C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection-diffusion systems, SIAM J. Numer. Anal., 35 (1998), 2440-2463.
[5] R. Guo and Y. Xu, Efficient solvers of discontinuous Galerkin discretization for the Cahn-Hilliard equations, J. Sci. Comp., 58 (2014), 380-408.
[6] C.A. Kennedy and M.H. Carpenter, Additive Runge-Kutta schemes for convection-diffusion-reaction equations, Appl. Num. Math., 44 (2003), 139-181.
[7] C.M. Klaij, M.H. van Raalte, H. van der Ven and J.J.W. van der Vegt, h-Multigrid for space-time discontinuous Galerkin discretizations of the compressible Navier-Stokes equations, J. Comput. Phys., 227 (2007), 1024-1045.
[8] H. Liu and J. Yan, A local discontinuous Galerkin method for the KdV equation with boundary effect, J. Comput. Phys., 215 (2006), 197-218.
[9] H. Liu and J. Yan, The direct discontinuous Galerkin (DDG) methods for diffusion problems, SIAM J. Numer. Anal., 47(2009), 675–698.
[10] W. Reed and T. Hill, Triangular mesh methods for the neutrontransport equation, La-ur-73-479, Los Alamos Scientific Laboratory, 1973.
[11] K. Shahbazi, D. J. Mavriplis and N.K. Burgess, Multigrid algorithms for high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations, J. Comput. Phys., 228 (2009), 7917-7940.
[12] J.J.W. van der Vegt and S. Rhebergen, hp-multigrid as smoother algorithm for higher order discontinuous Galerkin discretizations of advection dominated flows. Part I. Multilevel Analysis, J. Comput. Phys., 231 (2012), 7537-7563.
[13] J.J.W. van der Vegt and S. Rhebergen, hp-multigrid as smoother algorithm for higher order discontinuous Galerkin discretizations of advection dominated flows. Part II. Optimization of the Runge-Kutta smoother, J. Comput. Phys., 231 (2012), 7564-7583.
[14] Y. Xia, Y. Xu and C.-W. Shu, Efficient time discretization for local discontinuous Galerkin methods, Discrete Contin. Dyn. Syst. Ser. B., 8 (2007), 677-693.
[15] Y. Xu and C.-W. Shu, Local discontinuous Galerkin methods for three classes of nonlinear wave equations, J. Comput. Math., 22 (2004), 250-274.
[16] Y. Xu and C.-W. Shu, Local discontinuous Galerkin methods for two classes of two dimensional nonlinear wave equations, Physica D, 208 (2005), 21-58.
[17] Y. Xu and C.-W. Shu, Local discontinuous Galerkin methods for high-order time-dependent partial differential equations, Commun. Comput. Phys., 7 (2010), 1-46.
[18] J. Yan and C.-W. Shu, A local discontinuous Galerkin method for KdV type equations, SIAM J. Numer. Anal., 40 (2002), 769-791.
[19] J. Yan and C.-W. Shu, Local discontinuous Galerkin methods for partial differential equations with higher order derivatives, J. Sci. Comput., 17 (2002), 27-42.