6th European Conference on Computational Mechanics (ECCM 6)
7th European Conference on Computational Fluid Dynamics (ECFD 7)
1115 June 2018, Glasgow, UK

A TWO-STAGE FOURTH ORDER TEMPORAL DISCRETIZATION OF GRP-DG METHODS FOR COMPRESSIBLE FLUID DYNAMICS I: ONE DIMENSIONAL CASES

YUE WANG[1] AND JIEQUAN LI[2]

[1] Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics
Huayuan Road, Haidian District, Beijing, 10094, P. R. China
wang_yue@iapcm.ac.cn

[2] Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics
Huayuan Road, Haidian District, Beijing, 10094, P. R. China
li_jiequan@iapcm.ac.cn

Key words: Two-stage, Discontinuous Galerkin, Generalized Riemann Problem

Abstract. A two-stage fourth order temporal discretization approach is developed for compressible fluid dynamics in the framework of discontinuous Galerkin (DG) methods. Different from the classical Runge-Kutta (RK) temporal discretization, we use the two-stage temporal discretization and take the generalized Riemann problem (GRP) solver to construct numerical fluxes for better resolution. It turns out that we just need to solve moments and reconstruct initial data twice at every time step so as to save the computational cost and decrease the memory footprint. Numerical results verified the performance of the proposed scheme.

1 INTRODUCTION

High order numerical methods are attractive in the simulation of complex fluid flows because of their potential in achieving better resolution and accuracy with small cell numbers, which often leads to save computational cost. As one of the popular high order schemes, the DG method has been well developed since Cockburn and Shu's fundamental contributions [7, 8, 9, 10, 30]. This method has many advantages over other methods such as the compactness, parallelization, feasibility to complex geometries and adaptivity, etc, as pointed out in [30], where a wide range of references about the DG method can be also found. The *temporal discretization* and related *solvers for numerical fluxes* are two fundamental ingredients in the DG method. Our idea will be introduced by analyzing the existing methods from these two aspects.

The temporal discretization methods for the DG methods usually can be divided into

two types: (i) The Runge-Kutta type and (ii) The Lax-Wendroff type. Traditional Runge-Kutta methods [7, 8, 9, 10] are the most popular since they are easy to implement and require low-storage. It can also be designed to be strong stability preserving (SSP). However, only Riemann solvers are applied to obtain *numerical fluxes with first order temporal accuracy*. The low-storage and TVD (total variation diminishing) Runge-Kutta methods require *many stages* to increase degrees of freedom resulting in *considerable communication overhead and increasing computational cost*. Furthermore explicit Runge-Kutta methods with positive SSP coefficients cannot be more than fourth-order accurate [26].

The Lax-Wendroff type DG methods [12, 25] have optimally low-storage, and contain minimal communication overhead for hyperbolic problems by updating unknowns at only one stage. They use approximate high order Riemann solvers (such as the ADER method [33]) to obtain high order numerical fluxes directly. However, one needs to have access to high order time derivatives of the unknowns for a Lax-Wendroff method. After producing the Jacobian of the flux function, the next time derivative produces the Hessian of the flux function. Further derivatives require tensors which grow vastly in size which will make the implementation more complicate, especially in higher dimensions.

In order to combine the advantages of the Runge-Kutta type and the Lax-Wendroff type methods, a *two-stage fourth order temporal discretization* is introduced to the DG method in this paper. This temporal discretization for an ordinary differential equation $y_t = L(y)$ is explained below.

Algorithm. *The two-stage fourth order algorithm*

(i) **Compute intermediate values at** $t_* = t_n + \frac{1}{2}\Delta t$

$$y^* = y^n + \frac{1}{2}\Delta t L(y^n) + \frac{1}{8}\Delta t^2 \frac{\partial}{\partial t} L(y^n), \tag{1}$$

(ii) **Evolution of the solution at a full step step**

$$y^{n+1} = y^n + \Delta t L(y^n) + \frac{1}{6}\Delta t^2 \left(\frac{\partial}{\partial t} L(y^n) + 2\frac{\partial}{\partial t} L(y^*) \right). \tag{2}$$

This discretization method belongs to *multi-derivative time integrators* which have a long history of development for ordinary differential equations [5, 19, 22, 23], and yet to date, only a small subset of these methods have been explored as a tool for solving partial differential equations, especially for hyperbolic conservation laws [11, 27, 32]. This large class of time integrators includes all popular multi-stage Runge-Kutta as well as single-step Lax-Wendroff methods.

1. Like Lax-Wendroff methods, multi-derivative integrators permit the evaluation of higher derivatives of the unknowns in order to decrease the memory footprint and communication overhead.

2. Like traditional Runge-Kutta methods, multi-derivative integrators admit the addition of extra stages, which introduce extra degrees of freedom that can be used to increase the order of accuracy or modify the region of absolute stability [6].

The solvers chosen for computing the numerical fluxes are related with the temporal discretization method. For the existing schemes (including the Runge-Kutta type and the Lax-Wendroff type), approximate solvers are mostly used which will partially lose some important physical information. In this paper we will choose the second order GRP solver. This choice is crucial since the governing equation is effectively used so that all physical information is incorporated into the numerical flux construction and the residual evaluation. In principle, the ingredient agrees with that in the Lax-Wendroff method and thus the resulting scheme produces physically admissible and stable approximate solutions. Of course, due to the presence of discontinuities in the solutions, the tracking technique of those discontinuities is adopted so that even very strong waves could be captured well. The GRP solver was first proposed in [1] and this first original version was based on the Lagragian framework. The systematic description with various applications was published in the monograph [2]. The direction Eulerian version was proposed for shallow water equations in [16], for gas dynamical system in [3], and general hyperbolic conservation laws in [4]. The GRP solver is also refined to illustrate how the thermodynamical variation is integrated into the design of high resolution methods for compressible fluid flows and demonstrate numerically the importance of thermodynamic effect in the resolution of strong waves [17]. More developments can be found in [15, 24, 18].

Finally let us remark the features of this new two-stage fourth order GRP-DG algorithm.

1. For multi-derivative time integrators, use at most two-derivatives is optimal and fundamental for hyperbolic conservation laws to retain portability, that is the two-stage fourth order case. Beyond two-derivatives, the many-derivative time integrators start to lose their portability.

2. Compared to the classical Runge-Kutta approach to achieve fourth order accuracy, we just spend two steps to complete one time level evolution so that at least computational time 50% are reduced in the process of the data reconstruction.

3. By analytically solving the generalized Riemann problems for the numerical fluxes, a close coupling between the temporal and spatial evolution is introduced so that the resolution can be improved. The following numerical example (Fig. 1) in our previous work [34] verifies our viewpoint. The second order GRP-DG method performs better than the second order Runge-Kutta DG method with exact Riemann solver.

This paper is organized in four sections. Besides the introduction section here, a two-stage fourth order GRP-DG method is formulated in Section 2 with the second order GRP solver. In Section 3, several numerical examples are provided to display the performance of the proposed schemes. Finally we present a conclusive discussion in Section 4. The details of the GRP solver can be found in the Appendix A.

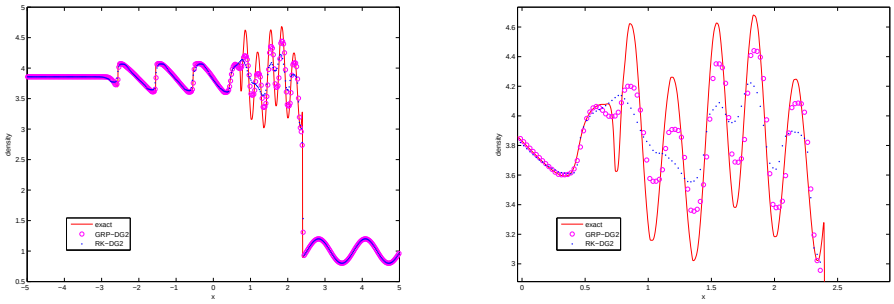


Fig. 1: The Shu-Osher problem. 400 cells are used. The second order GRP-DG method compares with the second order Runge-Kutta DG method with exact Riemann solver.

2 The two-stage fourth order GRP-DG methods for hyperbolic conservation laws

In this section we present how to implement the two-stage temporal discretization for the DG method. We will focus on the one-dimensional case in this paper and the two-dimensional case will be introduced in coming days.

The one-dimensional case of hyperbolic conservation laws reads

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = 0. \tag{3}$$

The computational interval $I = [a, b]$ is divided into sub-cells I_j, $j = 1, \cdots, N$. Denote the space of polynomials of degree at most k over I_j by $P_j^k(x)$, $x \in I_j$, and

$$V_I^k = \left\{ \mathbf{v}_h \in L^\infty(I); \mathbf{v}_h|_{I_j} \in P^k(I_j), j = 1, \cdots, N \right\}. \tag{4}$$

As a fourth order DG scheme is considered in this paper, k is chosen to be 3. Then we define an approximate solution $\mathbf{u}_h(x, t)$ in I_j,

$$\mathbf{u}_h(x, t) = \sum_{\ell=0}^{k} \mathbf{C}_j^\ell(t)\phi_\ell^j(x), \quad x \in I_j, \tag{5}$$

where $\{\phi_\ell^j\}$ is the Legendre orthogonal basis of P_j^k in I_j and normalized so that $\phi_\ell^j(x_{j+\frac{1}{2}}) = 1$, $\phi_\ell^j(x_{j-\frac{1}{2}}) = (-1)^\ell$. Those basis functions are taken for P_j^k as

$$\begin{aligned} &\phi_0^j = 1, \quad \phi_1^j(x) = \frac{2}{\Delta x_j}(x - x_j), \quad \phi_2^j(x) = \frac{6}{\Delta x_j^2}(x - x_j)^2 - \frac{1}{2}, \\ &\phi_3^j(x) = \frac{20}{\Delta x_j^3}(x - x_j)^3 - \frac{3}{\Delta x_j}(x - x_j), ... \end{aligned} \tag{6}$$

Then we plug (5) into the weak form of (3) over I_j, by taking the test functions as $\phi_\ell^j(x)$, to obtain

$$\begin{aligned} \frac{d}{dt}\mathbf{C}_j^\ell(t) = \; & -\frac{\alpha_\ell}{\Delta x_j}(\hat{\mathbf{f}}_{j+\frac{1}{2}}\phi_\ell^j(x_{j+\frac{1}{2}}) - \hat{\mathbf{f}}_{j-\frac{1}{2}}\phi_\ell^j(x_{j-\frac{1}{2}})) \\ & + \frac{\alpha_\ell}{\Delta x_j}\int_{I_j} \mathbf{f}(\mathbf{u}_h(x, t))\frac{d}{dx}\phi_\ell^j(x)dx = 0, \quad \ell = 0, \cdots, k, \end{aligned} \tag{7}$$

where $\hat{\mathbf{f}}_{j+\frac{1}{2}}$ is the numerical flux through the cell boundary $x = x_{j+\frac{1}{2}}$ at time t, and α_ℓ are

$$\alpha_\ell = \frac{\Delta x_j}{\int_{I_j}[\phi_\ell^j(x)]^2 dx}, \quad \alpha_0 = 1, \quad \alpha_1 = 3, \quad \alpha_2 = 5, \quad \alpha_3 = 7, \quad \cdots. \tag{8}$$

It turns out that we have the system for $\mathbf{C}_j^\ell$, $\ell = 0, 1, \cdots, k$,

$$\begin{aligned}
&\frac{d\mathbf{C}_j^0(t)}{dt} = -\frac{1}{\Delta x_j}[\hat{\mathbf{f}}_{j+\frac{1}{2}} - \hat{\mathbf{f}}_{j-\frac{1}{2}}],\\
&\frac{d\mathbf{C}_j^1(t)}{dt} = -\frac{3}{\Delta x_j}[\hat{\mathbf{f}}_{j+\frac{1}{2}} + \hat{\mathbf{f}}_{j-\frac{1}{2}}] + \frac{6}{\Delta x_j^2}\int_{I_j}\mathbf{f}(\mathbf{u}_h(x,t))dx,\\
&\frac{d\mathbf{C}_j^2(t)}{dt} = -\frac{5}{\Delta x_j}[\hat{\mathbf{f}}_{j+\frac{1}{2}} - \hat{\mathbf{f}}_{j-\frac{1}{2}}] + \frac{60}{\Delta x_j^3}\int_{I_j}\mathbf{f}(\mathbf{u}_h(x,t))(x-x_j)dx,\\
&\frac{d\mathbf{C}_j^3(t)}{dt} = -\frac{7}{\Delta x_j}[\hat{\mathbf{f}}_{j+\frac{1}{2}} + \hat{\mathbf{f}}_{j-\frac{1}{2}}] + \frac{21}{\Delta x_j^4}\int_{I_j}\mathbf{f}(\mathbf{u}_h(x,t))(20(x-x_j)^2 - \Delta x_j^2)dx,\\
&\cdots
\end{aligned} \tag{9}$$

The integrals in (9) can be evaluated using the Gauss-Lobatto quadrature since the end-point values of integrands are used. For example, we approximate,

$$\frac{1}{\Delta x_j}\int_{I_j}\mathbf{f}(\mathbf{u}_h(x,t))dx = \frac{1}{k(k+1)}(\hat{\mathbf{f}}_{j-\frac{1}{2}}(t) + \hat{\mathbf{f}}_{j+\frac{1}{2}}(t)) + \sum_{p=1}^{k}\chi_p\mathbf{f}(\mathbf{u}_h(x_p,t)), \tag{10}$$

where x_p is the $(p-1)$st zero of the Legendre polynomial $L'_{k-1}(x)$ and χ_p is the weight

$$\chi_p = \frac{\Delta x_i}{k(k+1)[L_{k-1}(x_p)]^2}, \quad x_p \neq x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}. \tag{11}$$

Thus the right-hand side of (9) can be regarded as functions of t so that (9) is an ODE system as long as the numerical flux $\hat{\mathbf{f}}_{j+\frac{1}{2}}$ can be defined and $\mathbf{u}_h(x_i,t)$ can be precised for interior points x_i. The latter is easy to determine because the solution is smooth so that

$$\frac{\partial \mathbf{u}_h}{\partial t}(x_i,t) = -\frac{\partial \mathbf{f}(u_h(x_i,t))}{\partial x}\bigg|_{x=x_i}, \quad x_i \neq x_{j-\frac{1}{2}}, \quad x_{j+\frac{1}{2}}. \tag{12}$$

It remains to determine $\hat{\mathbf{f}}_{j+\frac{1}{2}}$. This is achieved through solving the generalized Riemann problem, subject to initial data

$$\mathbf{u}(x,t_n) = \begin{cases} \mathbf{u}_h(x,t_n), & x \in I_j,\\ \mathbf{u}_h(x,t_n), & x \in I_{j+1}. \end{cases} \tag{13}$$

Then we obtain

$$\mathbf{u}_{j+\frac{1}{2}}^n := \lim_{t\to t_n+0}\mathbf{u}(x_{j+\frac{1}{2}},t), \qquad \left(\frac{\partial \mathbf{u}}{\partial t}\right)_{j+\frac{1}{2}}^n := \lim_{t\to t_n+0}\frac{\partial}{\partial t}\mathbf{u}(x_{j+\frac{1}{2}},t). \tag{14}$$

The procedure obtaining the values in (14) is named *the GRP solver* [3]. We refer to Appendix A for details. Intrinsically, the temporal derivative $(\partial \mathbf{u}/\partial t)^n_{j+\frac{1}{2}}$ is replaced by the spatial derivative at time $t = t_n$ using the governing equation (3),

$$\left(\frac{\partial \mathbf{u}}{\partial t}\right)^n_{j+\frac{1}{2}} = -\lim_{t\to t_n+0}\frac{\partial}{\partial x}\mathbf{f}(\mathbf{u}(x_{j+\frac{1}{2}},t)). \tag{15}$$

where the spatial derivative should be taken upwind. This approach is called the Lax-Wendroff approach numerically or the Cauchy-Kovalevskaya approach in the context of PDE theory.

Once the instantaneous values in (14) are available, we can define the following values within second order accuracy for $0 < \alpha < 1$,

$$\begin{aligned}
&\hat{\mathbf{f}}_{j+\frac{1}{2}}(t_n+\alpha\Delta t) = \mathbf{f}(\mathbf{u}^{t_n+\alpha\Delta t}_{j+\frac{1}{2}}), \quad \mathbf{u}^{t_n+\alpha\Delta t}_{j+\frac{1}{2}} := \mathbf{u}^n_{j+\frac{1}{2}} + \left(\frac{\partial \mathbf{u}}{\partial t}\right)^n_{j+\frac{1}{2}}\alpha\Delta t,\\
&\mathbf{f}(u_h(x_i,t_n+\alpha\Delta t)) = \mathbf{f}(\mathbf{u}^{t_n+\alpha\Delta t}_{h,i}), \quad \mathbf{u}^{t_n+\alpha\Delta t}_{h,i} = \mathbf{u}_h(x_i,t_n) + \frac{\partial \mathbf{u}_h}{\partial t}(x_i,t_n)\alpha\Delta t
\end{aligned} \tag{16}$$

Similarly, we can define the values after time level $t = t_*$. Thus we can implement our two-stage algorithm for (9), for which we denote symbolically

$$\frac{d\mathbf{C}_j(t)}{dt} = \mathbf{L}_j(\mathbf{u}_h(t),t), \tag{17}$$

and by $\mathbf{u}_h(t)$ all possible values involved in (9) at time t.

Algorithm-1D. Two-stage GRP-DG for 1-D hyperbolic conservation laws.

Step 1. Computation of intermediate values. Update the intermediate values $\mathbf{u}_h(x,t_*)$ at $t = t_n + \frac{1}{2}\Delta t$ in the following,

$$\mathbf{C}^*_j = \mathbf{C}_j(t_n) + \frac{\Delta t}{2}\mathbf{L}_j(\mathbf{u}_h(t_n),t_n) + \frac{\Delta t^2}{8}\frac{\partial}{\partial t}\mathbf{L}_j(\mathbf{u}_h(t_n),t_n). \tag{18}$$

Then we reconstruct $\mathbf{u}_h(x,t_*)$ and subsequently obtain values $\mathbf{u}^*_{j+\frac{1}{2}}$, $(\frac{\partial \mathbf{u}}{\partial t})^*_{j+\frac{1}{2}}$ and $\frac{\partial \mathbf{u}_h}{\partial t}(x_i,t_*)$.

Step 2. Advancing of solutions from t_n to t_{n+1}. Advance the solution to the time level $t = t_n + \Delta t$,

$$\mathbf{C}_j(t_{n+1}) = \mathbf{C}_j(t_n) + \Delta t\mathbf{L}_j(\mathbf{u}_h(t_n),t_n) + \frac{\Delta t}{6}\left[\frac{\partial}{\partial t}\mathbf{L}_j(\mathbf{u}_h(t_n),t_n) + 2\frac{\partial}{\partial t}\mathbf{L}_j(\mathbf{u}_h(t_*,t_*))\right]. \tag{19}$$

Thus $\mathbf{u}_h(x,t_{n+1})$ is reconstructed and subsequently $\mathbf{u}^{n+1}_{j+\frac{1}{2}}$, $(\frac{\partial \mathbf{u}}{\partial t})^{n+1}_{j+\frac{1}{2}}$ and $\frac{\partial \mathbf{u}_h}{\partial t}(x_i,t_{n+1})$ are obtained. Back to Step 1 until the time is up.

This is exactly a two-stage method. At each stage, we need to modify $\mathbf{C}_j^\ell$ in order to avoid oscillations. This modification is achieved using a shock-capturing limiter proposed by [21].

Remark We stick to the utilization of the time derivative $(\partial \mathbf{u}/\partial t)^n_{j+\frac{1}{2}}$, which is one of central points in our algorithm. Indeed, the fully explicit form of (3) is,

$$\bar{\mathbf{u}}_j^{n+1} = \overline{\mathbf{u}}_j^n - \frac{\Delta t}{\Delta x_j}\left[\frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \mathbf{f}(\mathbf{u}(x_{j+\frac{1}{2}},t))dt - \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \mathbf{f}(\mathbf{u}(x_{j-\frac{1}{2}},t))dt\right]. \tag{20}$$

It is crucial to approximate the flux in the sense that

$$\text{Numerical flux at } x_{j+\frac{1}{2}} - \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \mathbf{f}(\mathbf{u}(x_{j+\frac{1}{2}},t))dt = \mathcal{O}(\Delta t^r), \quad r>0. \tag{21}$$

Many algorithms approximate the flux with error measured by $\Delta \mathbf{u}$, the jump across the interface,

$$\text{Numerical flux at } x_{j+\frac{1}{2}} - \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \mathbf{f}(\mathbf{u}(x_{j+\frac{1}{2}},t))dt = \mathcal{O}(\|\Delta u\|^r). \tag{22}$$

which is not proportional to the mesh size Δx_j or the time step length Δt when the jump is large,

$$\|\Delta \mathbf{u}\| \not\approx \mathcal{O}(\Delta x_j). \tag{23}$$

It turns out that there is a large discrepancy when strong discontinuities are present in solutions. In order to overcome this difficulty, we have to solve the associated generalized Riemann problem (GRP) analytically and derive the value $(\partial \mathbf{u}/\partial t)^n_{j+\frac{1}{2}}$ and subsequently $(\partial \mathbf{u}/\partial t)^*_{j+\frac{1}{2}}$.

3 Numerical Examples

In this section we provide several examples to validate the performance of the proposed approach. The examples include linear and nonlinear scalar conservation laws and 1-D Euler equations. The order of accuracy will be tested. All results are obtained with CFL number $\nu = 0.12$. Let us compare it with the data from [27] (Table 1).

CFL number	ν	ν_{max}
SSP-RK3-DG (3 stages)	0.125	0.13
SSP-RK4-DG (10 stages)	0.44	0.45
TVRK4-DG (2 stages)	0.08	0.085
2stage-GRPDG	0.12	0.1225

Table 1: The comparison of the CFL numbers for different versions of DG methods

The SSP-RK3-DG method is the optimal third order SSP method developed by Shu and Osher [28] and described by Gottlieb and Shu [14]. SSP-RK4-DG is a fourth-order, low-storage method with ten stages developed by Ketcheson [20]. TVRK4-DG is the two-stage DG method with approximate Riemann solvers (TVRK4) introduced by Seal [27]. The maximum allowable CFL number, ν_{max} is near the maximum possible stable time step that each method permits for fourth-order spatial accuracy. SSP-RK4 has a much higher CFL limit when compared to either TDRK4 or SSP-RK3 or 2stage-GRPDG because it incorporates many more stages, but each stage requires expensive applications of limiters. Although TVRK4-DG shares the same temporal discretization with 2stage-GRPDG, it is interesting that using an exact high order Riemann solver will help us to loose the CFL constraint.

3.1 Scalar conservation laws

Example 1. The first example is a linear equation with a periodic boundary condition,

$$u_t + u_x = 0, \qquad u(x,0) = \sin(\pi x). \tag{24}$$

The solution is computed over the interval $[-1,1]$. This example is applied to test that our scheme can reach expected fourth order accuracy (see Table 2).

N	L^1 error	order	L^∞ error	order
20	$0.2176e-04$	-	$0.1680e-04$	-
40	$0.1348e-05$	4.013	$0.1045e-05$	4.007
80	$0.8352e-07$	4.012	$0.6530e-07$	4.001
160	$0.5203e-08$	4.005	$0.4079e-08$	4.001
320	$0.3246e-09$	4.002	$0.2547e-09$	4.001
640	$0.2027e-10$	4.001	$0.1592e-10$	4.000

Table 2: Numerical Accuracy Test for Example 1

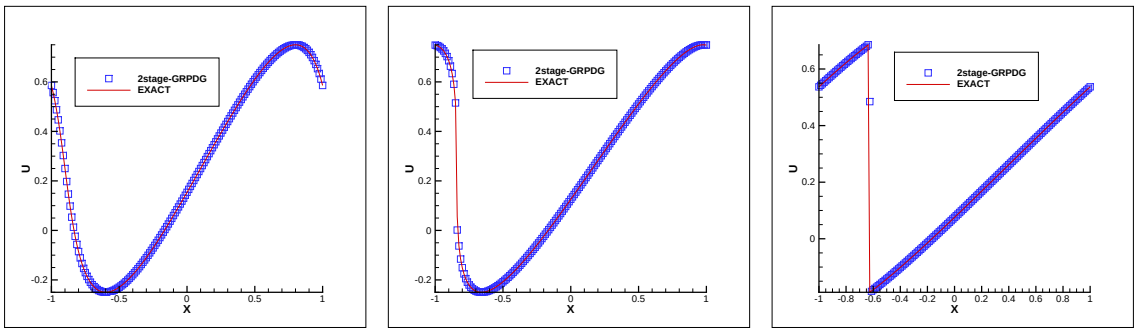
Example 2. The second example is taken for the Burgers equation with a periodic boundary condition [7],

$$u_t + (u^2/2)_x = 0, \quad u(x,0) = \frac{1}{4} + \frac{1}{2}\sin(\pi x). \tag{25}$$

The solution is smooth up to the time $t = 2/\pi$ and develops a shock that moves to interact with a rarefaction. Numerical examples at $t = 0.4$ [Fig. 2(a)], $t = 2/\pi$ [Fig. 2(b)] and $t = 1.5$ [Fig. 2(c)] are compared with the exact solutions by using 160 cells. Table 3 shows the L^1 and L^∞ errors and numerical order of the new scheme at $t = 0.4$. A high order limiter is only applied for the cases $t = 2/\pi$ and $t = 1.5$.

N	L^1 error	order	L^∞ error	order
20	$0.8743D-05$	-	$0.3355D-04$	-
40	$0.6102D-06$	3.841	$0.2093D-05$	4.002
80	$0.3886D-07$	3.973	$0.1639D-06$	3.675
160	$0.2449D-08$	3.988	$0.1243D-07$	3.721
320	$0.1563D-09$	3.970	$0.8796D-09$	3.821
640	$0.9914D-11$	3.979	$0.6024D-10$	3.868

Table 3: Numerical Accuracy Test for Example 2



(a) $t = 0.4$ (b) $t = 2/\pi$ (c) $t = 1.5$

Fig. 2: The numerical results for the Burgers equation at different times.

3.2 One-dimensional Euler equations

In this part we provide several examples for compressible Euler equations,

$$\mathbf{u} = (\rho, \rho v, E)^\top, \qquad \mathbf{f}(\mathbf{u}) = (\rho v, \rho v^2 + p, v(E+p))^\top, \tag{26}$$

where ρ is the density, v is the velocity, p is the pressure and $E = \rho(\frac{1}{2}v^2 + e)$ is the total energy, $e = \frac{p}{(\gamma-1)\rho}$ is the internal energy for polytropic gases. $\gamma = 1.4$ for all cases.

Example 3. Smooth problem In order to varify the numerical accuracy of the present high-order scheme for the Euler equations, we check the numerical results of the smooth problem whose initial data is

$$\begin{bmatrix} \rho \\ v \\ p \end{bmatrix} = \begin{bmatrix} 1 + 0.2\sin(x) \\ 1 \\ 1 \end{bmatrix}. \tag{27}$$

From Table 4 we could see that the present scheme reaches the designed order.

N	L^1 error	order	L^∞ error	order
20	$0.1486D-05$	-	$0.34420D-06$	-
40	$0.9769D-07$	3.927	$0.2347D-07$	3.874
80	$0.6102D-08$	4.001	$0.1497D-08$	3.970
160	$0.4006D-09$	3.929	$0.9925D-10$	3.915
320	$0.2443D-10$	4.035	$0.6080D-11$	4.029
640	$0.1529D-11$	3.999	$0.3813D-12$	3.995

Table 4: Numerical Accuracy Test for Example 3 at $t = 2\pi$

Example 4. The Sod problem This example was proposed in [31] to model a shock tube problem. The initial data is take as

$$\begin{aligned} &(\rho_\ell, v_\ell, p_\ell) = (1.0, 0.0, 1.0), && \text{for } 0 < x < 0.5, \\ &(\rho_r, v_r, p_r) = (0.125, 0.0, 0.1), && \text{for } 1 > x \geq 0.5. \end{aligned} \tag{28}$$

100 cells are applied to show the numerical results compared with the exact solution (Fig 3).

Example 5. The Shu-Osher problem This example was proposed in [29] to model shock-turbulence interactions. The initial data is take as

$$\begin{aligned} &(\rho_\ell, v_\ell, p_\ell) = (3.857143, 2.629369, 10.333333), && \text{for } x < -4, \\ &(\rho_r, v_r, p_r) = (1 + 0.2\sin(5x), 0, 1), && \text{for } x \geq -4. \end{aligned} \tag{29}$$

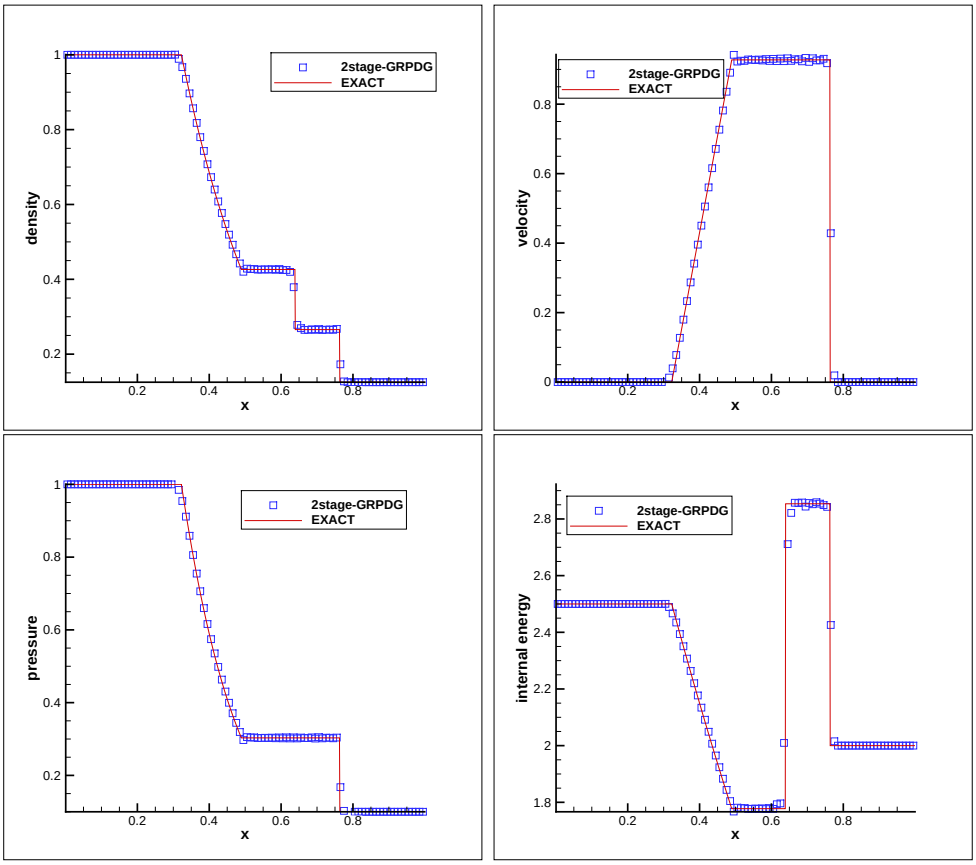


Fig. 3: The Sod problem: 100 cells are used

We can find out that results computed by 200 cells is comparable with the one computed by 300 cells which is enough to describe the detailed structures (Fig 4).

Example 6. The Woodward-Colella problem This test is the Woodward and Colella problem. The initial conditions in the present computation are the following:

$$\begin{aligned} &(\rho_\ell, v_\ell, p_\ell) = (1.0, 0.0, 1000.0), && \text{for } 0 < x < 0.1, \\ &(\rho_m, v_m, p_m) = (1.0, 0.0, 0.01), && \text{for } 0.1 < x < 0.9, \\ &(\rho_r, v_r, p_r) = (1.0, 0.0, 100.0), && \text{for } 0.9 < x < 1. \end{aligned} \tag{30}$$

600 cells are used to show the numerical results for the present scheme compared with the results by using 3200 cells (Fig 5).

4 Conclusion

This paper proposes a two-stage fourth order accurate temporal discretization of the DG method for hyperbolic conservation laws based on the GRP solver. The particular applications are given for compressible fluid dynamics. A number of numerical examples

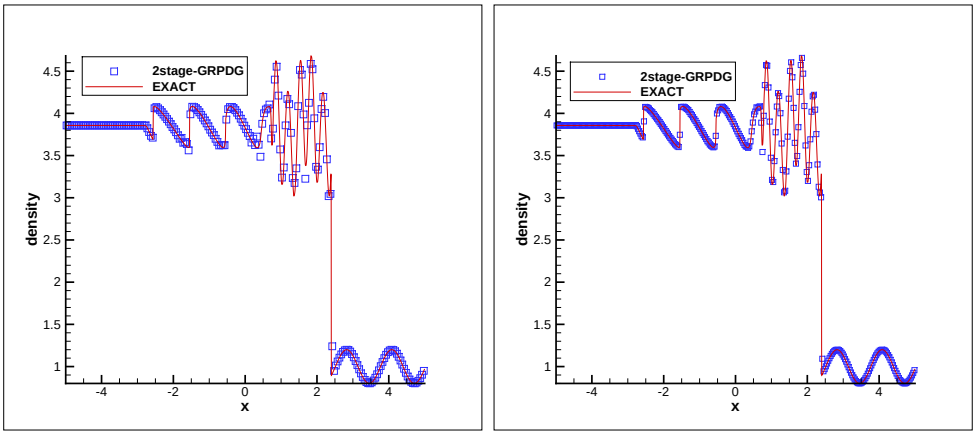


Fig. 4: The Shu-Osher problem: 200 (left) and 300 (right) cells are used

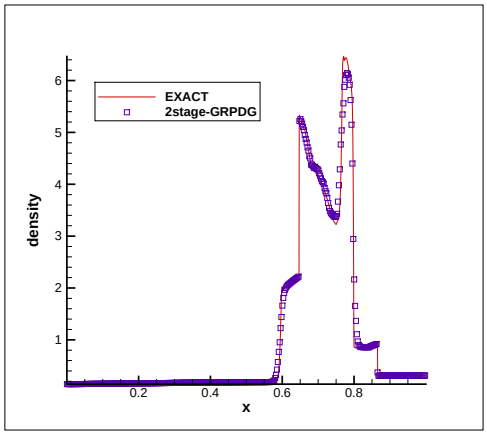


Fig. 5: The Woodward-Colella problem: 600 cells are used

are provided to validate the accuracy of the scheme and its computational performance for complex flow problems.

The resulting two-stage fourth order GRP-DG scheme is different with the existing DG schemes. A two-stage time discretization which belongs to the multi-derivative integrators combines the advantages of the Runge-Kutta and the Lax-Wendroff methods. It reduces the reconstruction compared with the Runge-Kutta method and avoid to compute large matrixes compared with the Lax-Wendroff method. Thus the computational cost is saved and the memory footprint and communication overhead are decreased. Furthermore, the GRP solver improves the resolution for capturing discontinuities as the governing equations are effectively used for the numerical flux reconstruction. It is interesting that the GRP solver also can help us to loose the CFL constraint. However, the performance of the proposed scheme for two-dimensional cases is needed to be studied.

A The GRP solver

This part includes the GRP solver just for completeness and readers' convenience. The details can be found in [3] for the Euler equations and [4] for general hyperbolic systems,

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = \mathbf{g}(\mathbf{u}, x), \tag{31}$$

where $\mathbf{g}(\mathbf{u}, x)$ is a source term. This paper only focuses on the homogeneous case, $\mathbf{g}(\mathbf{u}, x) \equiv 0$.

A.1 1-D GRP

The 1-D GRP solver assumes that the initial data consist of two pieces of polynomials,

$$\mathbf{u}(x,0) = \begin{cases} \mathbf{u}_-(x), & x < 0, \\ \mathbf{u}_+(x), & x > 0, \end{cases} \tag{32}$$

where $\mathbf{u}_\pm(x)$ are two polynomials with limiting states

$$\begin{aligned} \mathbf{u}_\ell &= \lim_{x\to 0-0} \mathbf{u}_-(x), \qquad \mathbf{u}_r = \lim_{x\to 0+0} \mathbf{u}_+(x); \\ \mathbf{u}'_\ell &= \lim_{x\to 0-0} \mathbf{u}'_-(x), \qquad \mathbf{u}'_r = \lim_{x\to 0+0} \mathbf{u}'_+(x). \end{aligned} \tag{33}$$

The GRP solver has two versions: (i) Acoustic version; (ii) Genuinely nonlinear version.

A.1.1 Acoustic GRP

The acoustic GRP deals with weak discontinuities or smooth flows and assumes that

$$\|\mathbf{u}_\ell - \mathbf{u}_r\| \ll 1. \tag{34}$$

However, we emphasize that the difference $\mathbf{u}'_\ell - \mathbf{u}'_r$ is not necessarily small. Then we denote by

$$\mathbf{u}_0 \approx \mathbf{u}_\ell \approx \mathbf{u}_r, \tag{35}$$

and linearize (3) around $\mathbf{u}_0$ as

$$\mathbf{u}_t + A(\mathbf{u}_0)\mathbf{u}_x = 0, \qquad A(\mathbf{u}_0) := \frac{\mathbf{f}(\mathbf{u}_0)}{\partial \mathbf{u}}. \tag{36}$$

Then the instantaneous time derivative of $\mathbf{u}$ is computed as,

$$\left(\frac{\partial \mathbf{u}}{\partial t}\right)_0 := \lim_{t\to 0+0} \frac{\partial \mathbf{u}}{\partial t}(0,t) = -[R\Lambda^+ R^{-1}\mathbf{u}'_\ell + R\Lambda^- R^{-1}\mathbf{u}'_r], \tag{37}$$

where $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_m)$. Here λ_i, $i = 1, \cdots, m$ are the eigenvalues of $A(\mathbf{u}_0)$, R is the (left) eigenmatrix of $A(\mathbf{u}_0)$, $\Lambda^+ = \mathrm{diag}(\max(\lambda_i, 0))$, $\Lambda^- = \mathrm{diag}(\min(\lambda_i, 0))$. The acoustic GRP is named as the G_1 scheme in the series of GRP papers.

A.1.2 Genuinely nonlinear GRP

As the jump at $x = 0$ is large, the acoustic GRP is not sufficient to resolve the resulting strong discontinuities. Any "rough" approximation is dangerous since the error is measured with the jump $\|\mathbf{u}_r - \mathbf{u}_r\|$, which is not proportional to the mesh size in the practical computation and results in large numerical discrepancy. Therefore, we have to analytically solve the associated generalized Riemann problem (31)-(32) as the "genuinely" nonlinear GRP version, which is named as the G_∞ GRP. This version is interpreted as the Lax-Wendrofff approach plus the tracking of strong discontinuities.

Here we include the resolution of GRP (31)-(32) for the Euler equations (26). The instantaneous value $\mathbf{u}_0$ is obtained by the Riemann solver and $(\partial \mathbf{u}/\partial t)_0$ is obtained by solving a pair of algebraic equations essentially,

$$\begin{aligned} a_\ell \left(\frac{\partial v}{\partial t}\right)_0 + b_\ell \left(\frac{\partial p}{\partial t}\right)_0 &= d_\ell, \\ a_r \left(\frac{\partial v}{\partial t}\right)_0 + b_r \left(\frac{\partial p}{\partial t}\right)_0 &= d_r, \end{aligned} \tag{38}$$

where the coefficients a_i, b_i, d_i, $i = 1, 2$, are given explicitly in terms of the initial data (32), and their formulae can be found in [3].

Since the variation of entropy s is precisely quantified, the instantaneous time derivative of the density is then obtained using the equation of state $p = p(\rho, s)$,

$$dp = c^2 d\rho + \frac{\partial p}{\partial s} ds. \tag{39}$$

Acknowledgements

Yue Wang is supported by NSFC (No. 11501040) and President Foundation of China Academy of Engineering Physics (YZJJLX2016009). Jiequan Li is supported by NSFC (No. 11771054).

REFERENCES

[1] M. Ben-Artzi, J. Falcovitz, A second-order Godunov-type scheme for compressible fluid dynamics, *J. Comput. Phys.*, (1984) **55**:1–32.

[2] M. Ben-Artzi, J. Falcovitz, *Generalized Riemann Problems in Computational Fluid Dynamics*, Cambridge University Press, 2003.

[3] M. Ben-Artzi, J. Li and G. Warnecke, A direct Eulerian GRP scheme for compressible fluid flows, *J. Comput. Phys.*, (2006) **218**:19–43.

[4] M. Ben-Artzi and J. Li, Hyperbolic conservation laws: Riemann invariants and the generalized Riemann problem, *Numer. Math.*, (2007) **106**:369–425.

[5] R. Chan, A. Tsai, On explicit two-derivative Runge-Kutta methods, *Numer. Algorithms*, (2010) **53(2-3)**:171–194.

[6] A. Christlieb, S. Gottlieb, Z. Grant, D. Seal, Explicit Strong Stability Preserving Multistage Two-Derivative Time-Stepping Schemes, *J. Sci. Comput.*, (2015) **185 (3)**:1–29.

[7] B. Cockburn and C. W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws II: General framework, *Math. Comput.*, (1989) **52**:411–435.

[8] B. Cockburn, S. Lin, and C. W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One dimensional systems, *J. Comput. Phys.*, (1989) **84**: 90–113.

[9] B. Cockburn, S. Hou, and C. W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case, *Math. Comput.*, (1990) **54**:545–581.

[10] B. Cockburn and C. W. Shu, The Runge-Kutta discontinuous Galerkin finite element method for conservation laws V: Multidimensional systems, *J. Comput. Phys.*, (1998) **141**: 199–224.

[11] Z. Du and J. Li, A Novel Two-Stage Fourth Order Temporal Discretization Based on the Lax-Wendroff Approach for Hyperbolic Conservation Laws, *SIAM J. Sci. Comput.*, (2016) **38 (5)**:3046–3069.

[12] M. Dumbser, C. Munz, Building blocks for arbitrary high order discontinuous Galerkin schemes, *J. Sci. Comput.*, (2006) **27(1-3)**:215–230.

[13] B. Einfeldt, C. Munz, P. Roe and B. Sjogreen, On Godunov-type methods near low densities, *J. Comput. Phys.*, (1991) **92 (2)**:273–295.

[14] S. Gottlieb, C. W. Shu, Total variation diminishing Runge-Kutta schemes, *Math. Comput.*, (1998) **67 (221)**:73–85.

[15] E. Han, J. Li, H. Tang, An adaptive GRP scheme for compressible fluid flows, *J. Comput. Phys.*, (2010) **229**:1448–1466.

[16] J. Li, G. Chen, The generalized Riemann problem method for the shallow water equations with bottom topography, *Int. J. Numer. Methods (Eng.)*, (2006) **65**:834–862.

[17] J. Li, Y. Wang, Thermodynamical Effects and High Resolution Methods for Compressible Fluid Flows, *Journal of Computational Physics*, (2017) **343**: 340–354.

[18] J. Li, Y. Zhang, The adaptive GRP scheme for compressible fluid flows over unstructured meshes, *J. Comput. Phys.*, (2013) **242 (6)**:367–386.

[19] K. Kastlunger, G. Wanner, Runge Kutta processes with multiple nodes, *Computing (Arch. Elektron. Rechnen)*, (1972) **9**:9–24.

[20] D. Ketcheson, Highly efficient strong stability-preserving Runge-Kutta methods with low-storage implementations, *SIAM J. Sci. Comput.*, (2008) **30(4)**:2113–2136.

[21] S. A. Moe, J. A. Rossmanith, D. C. Seal, A Simple and Effective High-Order Shock-Capturing Limiter for Discontinuous Galerkin Methods, *Mathematics*, (2015) **6 (3)**:57–83.

[22] N. Obreschkoff, Neue Quadraturformeln, *Abh. Preuss. Akad. Wiss. Math.-Nat. Kl.*, 1940.

[23] H. Ono, T. Yoshida, Two-stage explicit Runge-Kutta type methods using derivatives, *Jpn. J. Ind. Appl. Math.*, (2004) **21 (3)**:361–374.

[24] J. Qi, Y. Wang, J. Li, Remapping-free Adaptive GRP Method for Multi-Fluid Flows I: One Dimensional Euler Equations, *Commun. Comput. Phys.*, (2014) **15 (4)**:1029–1044.

[25] J. Qiu, M. Dumbser, C. W. Shu, The discontinuous Galerkin method with Lax-Wendroff type time discretizations, *Comput. Methods Appl. Mech. Eng.*, (2005) **194 (42-44)**:4528–4543.

[26] S. Ruuth, R. Spiteri,Two barriers on strong-stability-preserving time discretization methods, *J. Sci. Comput.*, (2002) **17**:211–220.

[27] D. C. Seal, Y. *Güclü*, A. J. Christlieb, High-Order Multiderivative Time Integrators for Hyperbolic Conservation Laws, *J. Sci. Comput.*, (2014) **60**:101–140.

[28] C. W. Shu, S. Osher, Efficient implementation of essentially nonoscillatory shock-capturing schemes, *J. Comput. Phys.*,(1988) **77 (2)**:439–471.

[29] C. W. Shu and S. Osher, Efficient implementation of essentially nonoscillatory shock-capturing schemes II, *J. Comput. Phys.*, (1989) **83**:32–78.

[30] C. W. Shu, Discontinuous Galerkin method for time dependent problems: Survey and recent developments, Recent Developments in Discontinuous Galerkin Finite Element Methods for Partial Differential Equations (2012 John H. Barrett Memorial Lectures), X. Feng, O. Karakashian and Y. Xing, editors, *The IMA Volumes in Mathematics and Its Applications*, 157 (2014), 25-62, Springer, Switzerland.

[31] G. A. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, *J. Comput. Phys.*, (1978) **27**:1–31.

[32] A. Tsai, R. Chan, S. Wang, Two-derivative Runge-Kutta methods for PDEs using a novel discretization approach, *Numer. Algor.*, (2014) **65**:687–703.

[33] E. F. Toro and V. A. Titarev, Derivative Riemann solvers for systems of conservation laws and ADER methods, *J. Comput. Phys.*, (2006) **212**:150–165.

[34] Y. Wang, S. Wang, Arbitrary high order discontinuous Galerkin schemes based on the GRP method for compressible Euler equations, *J. Comput. Phys.*, (2015) **298**:113–124.