



A new ADER discontinuous Galerkin method based on differential transformation procedure for hyperbolic conservation laws

Yingjuan Zhang¹ · Gang Li¹ · Shouguo Qian¹ · Jinmei Gao¹

Received: 9 September 2020 / Revised: 6 April 2021 / Accepted: 6 May 2021 /
Published online: 15 May 2021

© SBMAC - Sociedade Brasileira de Matemática Aplicada e Computacional 2021

Abstract

This article develops a new discontinuous Galerkin (DG) method with the one-stage arbitrary derivatives in time and space approach to solve one-dimensional hyperbolic conservation laws. This method employs the differential transformation procedure instead of the Cauchy–Kowalewski procedure to recursively express the spatiotemporal expansion coefficients of the solution through the low-order spatial expansion coefficients. The proposed method is free of solving generalized Riemann problems at inter-cells. Compared with the Runge–Kutta DG methods, the current method needs less computer memory due to no intermediate stages. In summary, this method is one step, one stage, fully discrete, and easily achieves arbitrary high-order accuracy in time and space. Extensive numerical results illustrate the good performances of the present method: high-order accuracy for smooth solutions, good resolution for discontinuous solutions and high efficiency.

Keywords Hyperbolic conservation laws · Discontinuous Galerkin method · ADER approach · Differential transformation procedure

Mathematics Subject Classification 65N30

Communicated by Abdellah Hadjad.

✉ Shouguo Qian
sgqian1976@163.com

Yingjuan Zhang
yingjuanzhang1997@163.com

Gang Li
gangli1978@163.com

Jinmei Gao
gjinmei1979@163.com

¹ School of Mathematics and Statistics, Qingdao University, Qingdao 266071, Shandong, People's Republic of China

1 Introduction

The hyperbolic conservation laws arise in many scientific as well as technological fields, such as the Euler equations in gas dynamics, shallow water equations, and so on. In this article, we are concerned with numerically solving hyperbolic conservation laws in one space dimension with the below form

$$U_t + F(U)_x = 0, \quad (1)$$

where U is the state vector; $F(U)$ is the physical flux. In recent years, researchers have paid more and more attention to the arbitrary high-order methods for hyperbolic conservation laws (Toro et al. 2001; Titarev and Toro 2002, 2005; Pedro et al. 2014).

The discontinuous Galerkin (DG) methods are popular to solve the hyperbolic conservation laws. The traditional DG methods by Cockburn and Shu (1989, 1998) and Cockburn et al. (1989, 1990) are semi-discrete methods, apply the multi-stage Runge–Kutta approach (Gottlieb et al. 2009) to obtain high-order temporal accuracy, and are known as the RKDG methods. A brief historic review as well as the latest developments can be found in Cockburn et al. (2000) and Shu (2016) and the references cited therein. The RKDG methods enjoy the advantage of simplicity, but need to construct the inter-cell numerical flux and to calculate volume integrations at each stage; therefore, the RKDG methods are very time consuming relatively.

Alternatively, Qiu et al. (2005) developed a DG method using the Lax–Wendroff (LWDG) approach for the temporal discretization; the resulting LWDG method is one stage and explicit. In addition, Dumbser and Munz (2005a, b) and Dumbser (2005) originally developed a new DG method with the one-stage ADER approach (Toro et al. 2001; Titarev and Toro 2002, 2005) for the temporal discretization. Subsequently, Dumbser and collaborators conducted a series of researches for this subject (Zanotti et al. 2015; Fambri et al. 2017, 2018; Rannabauer et al. 2018). The key ingredient of the DG methods (Zanotti et al. 2015; Fambri et al. 2017, 2018; Rannabauer et al. 2018) is to construct high-order inter-cell numerical flux. The inter-cell state is expressed in a form of temporal Taylor series expansion, where the temporal derivatives are described by the spatial derivatives using the governing PDE repeatedly. This procedure is known as the Cauchy–Kowalewski procedure [also termed as the Lax–Wendroff procedure in the literature Harten (1987)]. In addition, one also need to solve generalized Riemann problems at inter-cells to obtain the state itself and its spatial derivatives. However, the Cauchy–Kowalewski procedure is very cumbersome especially for high-dimensional problems. Therefore, the replacement or simplification of the Cauchy–Kowalewski procedure is very welcome. It is noteworthy that Dumbser and Munz (2006) have proposed an efficient algorithm for the Cauchy–Kowalewski procedure. More recently, Dumbser et al. (2014) and Duan and Tang (2020) have adopted the local continuous space-time Galerkin predictor approach (Dumbser et al. 2008) to replace the Cauchy–Kowalewski procedure under the framework of DG methods.

The key objective of this article is to develop a new DG method with the ADER approach for the temporal discretization, which is called as the ADER-DG method accordingly. The method here applies the differential transformation procedure Ayaz (2003, 2004) and Kurnaza et al. (2005) instead of the Cauchy–Kowalewski procedure as in Zanotti et al. (2015), Fambri et al. (2017, 2018) and Rannabauer et al. (2018) to express the spatiotemporal expansion coefficients of the solution through the low-order spatial expansion coefficients and enables us to avoid solving the generalized Riemann problem at inter-cells.

In comparison with the traditional RKDG methods (Cockburn and Shu 1989, 1998; Cockburn et al. 1989, 1990), the proposed method is one stage and fully discrete, which is welcome

in many practical applications due to its simplicity and the low computer storage. Moreover, the one-stage reformulation greatly reduces the frequency of inter-process communications, and is desirable for the parallel computing. The current method is free of solving generalized Riemann problems at inter-cells, compared with the existing methods (Zanotti et al. 2015; Fambri et al. 2017, 2018; Rannabauer et al. 2018), which apply the Cauchy–Kowalewski procedure. The differential transformation procedure is more concise and more efficient than the Cauchy–Kowalewski procedure (Norman and Finkel 2012). The space–time polynomial representation of the solution on each space-time control volume leads to precise calculations for the volume integral and the numerical flux and avoids the use of costly numerical quadrature rule.

This article is organized as follows. In Sect. 2, we present a new ADER-DG method with differential transformation procedure. Section 3 contains extensive examples to illustrate the performances of the current method. Some conclusions are given in Sect. 4.

2 Construction of ADER-DG method

2.1 Mesh and solution space

We first divide the spatial domain of interest with cells $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$, for $j = 1, 2, \dots, N$, and we set $x_j = \frac{1}{2}(x_{j-\frac{1}{2}} + x_{j+\frac{1}{2}})$ and $\tau_j = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$ as the mesh center and the mesh size, respectively. The notation $\tau = \max_{1 \leq j \leq N} \tau_j$ stands for the Maximum mesh size and the notation $\Omega_j = I_j \times [t^n, t^{n+1}]$ represents the space-time control volume with $[t^n, t^{n+1}]$ being the time interval.

Herein, we take the approximation solution space as follows

$$V_\tau^k = \left\{ \varphi(x, t) : \varphi(x, t)|_{\Omega_j} \in P^k(\Omega_j) \right\},$$

where $P^k(\Omega_j)$ denotes the set of space-time polynomials of degree up to k on Ω_j .

2.2 General formulation of ADER-DG method

Multiplication of (1) with an arbitrary spatial test function $\phi(x)$ and using the integration by parts on Ω_j yield the following weak form

$$\begin{aligned} & \int_{I_j} U(x, t^{n+1}) \phi(x) dx - \int_{I_j} U(x, t^n) \phi(x) dx \\ & + \int_{t^n}^{t^{n+1}} F\left(x_{j+\frac{1}{2}}, t\right) dt \cdot \phi\left(x_{j+\frac{1}{2}}^-\right) - \int_{t^n}^{t^{n+1}} F\left(x_{j-\frac{1}{2}}, t\right) dt \cdot \phi\left(x_{j-\frac{1}{2}}^+\right) - \iint_{\Omega_j} F(x, t) \phi_x dx dt \\ & = 0, \quad \text{for } j = 1, 2, \dots, N. \end{aligned}$$

Here, we take the physical flux as a binary function with $F(x, t) = F(U(x, t))$ for the sake of easy presentation.

The fully discrete ADER-DG method for solving (1) is defined as follows: for all test functions $\phi(x) = (x - x_j)^{k_x}$ with $k_x = 0, 1, \dots, k$ from V_τ^k , the solution $U_\tau(x, t) \in V_\tau^k$ satisfies the following equality

$$\begin{aligned}
 & \int_{I_j} U_\tau(x, t^{n+1}) \phi(x) \, dx \\
 &= \int_{I_j} U_\tau(x, t^n) \phi(x) \, dx - \widehat{F}_{j+\frac{1}{2}} \cdot \phi\left(x_{j+\frac{1}{2}}^-\right) + \widehat{F}_{j-\frac{1}{2}} \cdot \phi\left(x_{j-\frac{1}{2}}^+\right) \\
 &+ \iint_{\Omega_j} F_\tau(x, t) \phi_x \, dx \, dt, \quad \text{for } j = 1, 2, \dots, N.
 \end{aligned} \tag{2}$$

Here, the numerical flux $\widehat{F}_{j+\frac{1}{2}}$ is used to approximate the time integration of the physical flux at inter-cell, i.e., $\int_{t^n}^{t^{n+1}} F\left(x_{j+\frac{1}{2}}, t\right) \, dt$.

Herein, on each space–time control volume Ω_j , we take $U_\tau(x, t)$ (the subscript j is ignored for the sake of simplicity) in a piecewise space–time polynomial form to approximate the unknown solution $U(x, t)$. At time t^n , we have $U_\tau(x, t^n)$ at hands as a linear combination of spatial polynomials

$$U_\tau(x, t^n) = \sum_{k_x=0}^k \widetilde{U}(k_x, 0)(x - x_j)^{k_x},$$

with $\widetilde{U}(k_x, 0)$ being the expansion coefficient in space. Our basic goal is to construct the following space–time polynomial

$$U_\tau(x, t) = \sum_{k_t=0}^k \sum_{k_x=0}^{k-k_t} \widetilde{U}(k_x, k_t)(x - x_j)^{k_x} (t - t^n)^{k_t} \in V_\tau^k,$$

on Ω_j in the whole time interval $[t^n, t^{n+1}]$. Here, $\widetilde{U}(k_x, k_t)$ denote the space–time-dependent degrees of freedom (DOF) on Ω_j . In the following, we will obtain $\widetilde{U}(k_x, k_t)$ from $\widetilde{U}(k_x, 0)$ at the old time level t^n repeatedly using the differential transformation procedure presented in Sect. 2.2.1.

The evolution from $U_\tau(x, t^n)$ to $U_\tau(x, t^{n+1})$ needs the evaluation of the right hand side of (2): a space integral, a space–time integral as well as the construction of numerical fluxes at intercells.

2.2.1 A brief review of differential transformation procedure

The key component of DG methods (Zanotti et al. 2015; Fambri et al. 2017, 2018; Rannabauer et al. 2018) is to repeatedly differentiate the PDE and to express the temporal derivatives through the spatial ones using the Cauchy–Kowalewski procedure. However, the Cauchy–Kowalewski procedure becomes more cumbersome especially for higher order derivatives due to chain rules. To reduce the cost of the Cauchy–Kowalewski procedure, Dumbser and Munz (2006) have proposed an efficient algorithm based on the Leibniz rule for two-dimensional Euler equations under the DG frame. More recently, Dumbser et al. (2014) and Duan and Tang (2020) have employed the local continuous space–time Galerkin predictor approach to replace the Cauchy–Kowalewski procedure (Dumbser et al. 2008) in the framework of DG methods.

Herein, we apply the differential transformation procedure to replace the Cauchy–Kowalewski procedure. The differential transformation procedure was originally proposed

Table 1 Transformed functions of some functions encountered in this article

Functional form	Transformed form
$q(x, t) = c \cdot u(x, t)$	$\tilde{q}(k_x, k_t) = c \cdot \tilde{u}(k_x, k_t)$
$q(x, t) = \partial_x u(x, t)$	$\tilde{q}(k_x, k_t) = (k_x + 1)\tilde{u}(k_x + 1, k_t)$
$q(x, t) = \partial_t u(x, t)$	$\tilde{q}(k_x, k_t) = (k_t + 1)\tilde{u}(k_x, k_t + 1)$
$q(x, t) = u(x, t)v(x, t)$	$\tilde{q}(k_x, k_t) = \sum_{r=0}^{k_x} \sum_{s=0}^{k_t} \tilde{u}(r, s)\tilde{v}(k_x - r, k_t - s)$
$q(x, t) = \frac{1}{u(x, t)}$	$\tilde{q}(0, 0) = \frac{1}{\tilde{u}(0, 0)}$ $\tilde{q}(k_x, k_t) = -\frac{1}{\tilde{u}(0, 0)} \sum_{r=0}^{k_x} \sum_{\substack{s=0 \\ s+r>0}}^{k_t} \tilde{u}(r, s)\tilde{q}(k_x - r, k_t - s)$

for nonlinear initial value problems in the electric circuit analysis (Pukhov 1982; Zhou 1986). Ayaz generalized this procedure to two-dimensional case (Ayaz 2003) and system case (Ayaz 2004; Kurnaza et al. 2005) conducted the extension to n -dimensional case. Recently, by means of this procedure, Norman and Finkel (2012) have developed a multi-moment finite volume scheme for one-dimensional balance laws.

The detailed definition of the differential transformation procedure is as follows. Suppose that a function $u(x, t)$ in cell I_j at time t^n is known, the differential transformation is defined as in Ayaz (2003, 2004) and Kurnaza et al. (2005)

$$\tilde{u}(k_x, k_t) = \frac{1}{k_x!k_t!} \left. \frac{\partial^{k_x+k_t} u(x, t)}{\partial x^{k_x} \partial t^{k_t}} \right|_{x=x_j, t=t^n}, \text{ with } u(x, t) = \sum_{k_t=0}^k \sum_{k_x=0}^{k-k_t} \tilde{u}(k_x, k_t)(x - x_j)^{k_x} (t - t^n)^{k_t}, \tag{3}$$

where $u(x, t)$ is the original function and $\tilde{u}(k_x, k_t)$ denotes the transformed function. In fact, the transformed function $\tilde{u}(k_x, k_t)$ denotes the expansion coefficient of the truncated Taylor series expansion. The transformed functions used in this article are shown in Table 1.

Subsequently, we exemplify the differential transformation procedure with the non-linear Burgers' equation

$$u_t + f(u)_x = 0, \tag{4}$$

where $f(u) = \frac{1}{2}u^2$ denotes the physical flux. Combining the rules in Table 1, we implement the differential transformation procedure at both ends of (4) and obtain the below equality for the transformed functions

$$(k_t + 1)\tilde{u}(k_x, k_t + 1) + (k_x + 1)\tilde{f}(k_x + 1, k_t) = 0,$$

which leads to the following recurrence formula

$$\tilde{u}(k_x, k_t + 1) = -\frac{k_x + 1}{k_t + 1} \tilde{f}(k_x + 1, k_t), \tag{5}$$

with

$$\tilde{f}(k_x, k_t) = \frac{1}{2} \sum_{r=0}^{k_x} \sum_{s=0}^{k_t} \tilde{u}(r, s)\tilde{u}(k_x - r, k_t - s).$$

We here take the physical flux $f(u)$ as a binary function with $f(x, t) = f(u(x, t))$. On basis of $\tilde{u}(k_x, 0)$, $k_x = 0, 1, \dots, k$ coming from $u_\tau(x, t^n) = \sum_{k_x=0}^k \tilde{u}(k_x, 0)(x - x_j)^{k_x}$ at time t^n

and applying the above recurrence relation (5) repeatedly, we obtain

$$\tilde{u}(k_x, k_t) \text{ and } \tilde{f}(k_x, k_t), \text{ for } k_t = 0, 1, \dots, k; k_x = 0, 1, \dots, k - k_t.$$

Next, we achieve

$$u_\tau(x, t) = \sum_{k_t=0}^k \sum_{k_x=0}^{k-k_t} \tilde{u}(k_x, k_t)(x - x_j)^{k_x}(t - t^n)^{k_t} \in V_\tau^k,$$

$$f_\tau(x, t) = \sum_{k_t=0}^k \sum_{k_x=0}^{k-k_t} \tilde{f}(k_x, k_t)(x - x_j)^{k_x}(t - t^n)^{k_t} \in V_\tau^k,$$

on the space–time control volume Ω_j . We present the detailed steps of the procedure for Burgers’ equation in Appendix A.

Remark 1 Actually, to implement the differential transformation procedure for a given function is to obtain its transformed functions namely the expansion coefficients of the Taylor series form. Moreover, the differential transformation procedure converts a PDE into a series of recurrence relations for expansion coefficients of the solution in a Taylor series form.

2.2.2 The differential transformation procedure for Euler equations

Let us deal with the Euler equations in the gas dynamics

$$\begin{cases} \rho_t + (\rho u)_x = 0, \\ (\rho u)_t + (\rho u^2 + p)_x = 0, \\ E_t + (u(E + p))_x = 0. \end{cases} \tag{6}$$

Here, ρ is the density, u is the velocity, E is the total energy, p is the pressure, which is related to the total energy by

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho u^2 \tag{7}$$

with $\gamma = 1.4$ as the specific heats ratio. We can rewrite the system (6) in the conservative form (1) with the following conservative variable and physical flux

$$U = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}, \quad F(U) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(E + p) \end{pmatrix}.$$

Suppose in cell I_j and at time $t = t^n$, the solution $U_\tau(x, t^n)$ are known in the form

$$U_\tau(x, t^n) = (\rho_\tau(x, t^n), (\rho u)_\tau(x, t^n), E_\tau(x, t^n))^T = \sum_{k_x=0}^k \tilde{U}(k_x, 0)(x - x_j)^{k_x}. \tag{8}$$

Then, using the Table 1, we implement the differential transformation procedure on both ends of system (1) in a componentwise manner; we obtain the following equality of the transformed functions in a vector form

$$(k_t + 1)\tilde{U}(k_x, k_t + 1) + (k_x + 1)\tilde{F}(k_x + 1, k_t) = 0, \tag{9}$$

with the below auxiliary variables

$$\begin{aligned} \tilde{F}(k_x, k_t) &= \begin{pmatrix} (\tilde{\rho}u)(k_x, k_t) \\ (\gamma - 1)\tilde{E}(k_x, k_t) + \frac{1}{2}(3 - \gamma)\tilde{G}_3(k_x, k_t) \\ \gamma\tilde{G}_5(k_x, k_t) - \frac{1}{2}(\gamma - 1)\tilde{G}_6(k_x, k_t) \end{pmatrix}, \\ \tilde{G}_1(k_x, k_t) &= -\frac{1}{\tilde{\rho}(0, 0)} \sum_{r=0}^{k_x} \sum_{s=0}^{k_t} \tilde{\rho}(r, s)\tilde{G}_1(k_x - r, k_t - s), \\ \tilde{G}_2(k_x, k_t) &= \sum_{r=0}^{k_x} \sum_{s=0}^{k_t} (\tilde{\rho}u)(r, s)(\tilde{\rho}u)(k_x - r, k_t - s), \\ \tilde{G}_3(k_x, k_t) &= \sum_{r=0}^{k_x} \sum_{s=0}^{k_t} \tilde{G}_2(r, s)\tilde{G}_1(k_x - r, k_t - s), \\ \tilde{G}_4(k_x, k_t) &= \sum_{r=0}^{k_x} \sum_{s=0}^{k_t} \tilde{\rho}u(r, s)\tilde{G}_1(k_x - r, k_t - s), \\ \tilde{G}_5(k_x, k_t) &= \sum_{r=0}^{k_x} \sum_{s=0}^{k_t} \tilde{G}_4(r, s)\tilde{E}(k_x - r, k_t - s), \\ \tilde{G}_6(k_x, k_t) &= \sum_{r=0}^{k_x} \sum_{s=0}^{k_t} \tilde{G}_4(r, s)\tilde{G}_3(k_x - r, k_t - s), \end{aligned}$$

where we ignore the subscript j for the clarity of presentation. Moreover, Eq. (9) leads to the following recurrence formula

$$\tilde{U}(k_x, k_t + 1) = -\frac{k_x + 1}{k_t + 1} \tilde{F}(k_x + 1, k_t). \tag{10}$$

Then, starting from

$$\tilde{U}(k_x, 0) = (\tilde{\rho}(k_x, 0), (\tilde{\rho}u)(k_x, 0), \tilde{E}(k_x, 0))^T, \text{ for } k_x = 0, 1, \dots, k,$$

we can circularly get

$$\tilde{U}(k_x, k_t) \text{ and } \tilde{F}(k_x, k_t), \text{ for } k_t = 0, 1, \dots, k; k_x = 0, 1, \dots, k - k_t,$$

by repeatedly using the recurrence formula (10). Subsequently, we obtain $U_\tau(x, t)$ and $F_\tau(x, t)$ as follows

$$\begin{aligned} U_\tau(x, t) &= \sum_{k_t=0}^k \sum_{k_x=0}^{k-k_t} \tilde{U}(k_x, k_t)(x - x_j)^{k_x} (t - t^n)^{k_t} \in V_\tau^k, \\ F_\tau(x, t) &= \sum_{k_t=0}^k \sum_{k_x=0}^{k-k_t} \tilde{F}(k_x, k_t)(x - x_j)^{k_x} (t - t^n)^{k_t} \in V_\tau^k, \end{aligned}$$

on each space–time control volume Ω_j , for $j = 1, 2, \dots, N$ to approximate the unknown conservative state $U(x, t)$ and the physical flux $F(x, t) = F(U(x, t))$. We present the specific steps in Appendix B.

In the above formulae (10), we apply the following relations

$$\rho u^2 + p = \rho u^2 + (\gamma - 1) \left(E - \frac{1}{2} \rho u^2 \right) = (\gamma - 1) E + \frac{1}{2} (3 - \gamma) \rho u^2,$$

$$u(E + p) = u \left(E + (\gamma - 1) \left(E - \frac{1}{2} \rho u^2 \right) \right) = \gamma u E - \frac{1}{2} (\gamma - 1) u (\rho u^2),$$

due to the equation of state (7).

Remark 2 The key function of the differential transformation procedure is to supply the temporal evolution, locally for each cell, of the existing solution $U_\tau(x, t^n)$ at time t^n .

Remark 3 The Cauchy–Kowalewski procedure uses the symbolic expansions of the PDE itself directly, which needs costly recomputations of many terms and leads to exponential increase in complexity. While, the differential transformation procedure is considerably cheaper than the Cauchy–Kowalewski procedure and achieves the same aim with a predictable polynomial complexity.

2.2.3 Construction of high order numerical fluxes

To construct high-order numerical flux in ADER schemes (Toro et al. 2001; Titarev and Toro 2002, 2005; Dumbser and Munz 2005a, b, 2006; Dumbser 2005), one first expand the inter-cell state in a time Taylor series form, where the temporal derivatives are expressed by the spatial derivatives using the Cauchy–Kowalewski procedure. To get the spatial derivatives, one also need to solve generalized Riemann problems at inter-cells.

After applying the differential transformation procedure in cell I_j at time t^n , we have $U_\tau(x, t)$ as well as $F_\tau(x, t)$ on Ω_j at hands, and then we adopt the simple and efficient Lax–Friedrichs flux

$$\widehat{F}_{j+\frac{1}{2}} = \frac{1}{2} \int_{t^n}^{t^{n+1}} \left(F_\tau \left(x_{j+\frac{1}{2}}^-, t \right) + F_\tau \left(x_{j+\frac{1}{2}}^+, t \right) - \alpha \left(U_\tau \left(x_{j+\frac{1}{2}}^+, t \right) - U_\tau \left(x_{j+\frac{1}{2}}^-, t \right) \right) \right) dt, \tag{11}$$

where α is an estimate of largest wave speed on the whole spatial domain.

Because $U_\tau(x, t)$ and $F_\tau(x, t)$ are all space–time polynomials on Ω_j , we can adopt precise calculations for the numerical flux in (11) in stead of resorting to the costly numerical quadrature rule.

In addition, for the space integral and the space–time integral in (2), we also apply the precise calculations and substitute them into (2), then obtain the fully discrete ADER-DG method (2).

Remark 4 The present construction means of the numerical fluxes is free of solving generalized Riemann problem at inter-cells.

2.3 Implementation details of ADER-DG method

Here, we summarize the proposed method within one time step $t^n \rightarrow t^{n+1}$ as follows:

- (i) Initially, we obtain $\widetilde{U}(k_x, 0)$, $k_x = 0, 1, \dots, k$, from $U(x, 0)$ in each cell I_j , for $j = 1, 2, \dots, N$.
- (ii) At time t^n , by the recursive steps (10), we acquire $\widetilde{U}(k_x, k_t)$ and $\widetilde{F}(k_x, k_t)$ on basis of $\widetilde{U}(k_x, 0)$, then obtain $U_\tau(x, t)$ and $F_\tau(x, t)$.
- (iii) Construct numerical fluxes $\widehat{F}_{j+\frac{1}{2}}$ using the formula (11).
- (iv) Obtain $U_\tau(x, t^{n+1})$ according to the one-step formula (2).

(v) Apply a slope limiter on $U_\tau(x, t^{n+1})$ when needed.

Remark 5 The one-dimensional method proposed here can be directly generalized to high-dimensional problems. To handle the multi-dimensional problems, we need to pay special attention on the differential transformation procedure. In practice, in comparison to the one-dimensional case, we need to add an extra layer on the program for the differential transformation procedures, see Appendices A and B. In addition, we also need to be careful when initializing.

3 Numerical results

In this section, we conduct classical examples to validate the performance of the current method. In the following computations, we take the polynomial of degree two and four (i.e., $k = 2$ and $k = 4$), and set the CFL constant as 0.18 and 0.1, respectively.

3.1 The scalar case

3.1.1 Linear advection equation

We first consider the linear advection equation as in (Jiang and Shu 1996a)

$$\begin{cases} u_t + u_x = 0, & -1 \leq x \leq 1, \\ u(x, 0) = u_0(x), \text{ periodic,} \end{cases}$$

where

$$u_0(x) = \begin{cases} \frac{1}{6} (\mathcal{G}(x, \beta, z - \delta) + \mathcal{G}(x, \beta, z + \delta) + 4\mathcal{G}(x, \beta, z)), & -0.8 \leq x \leq -0.6, \\ 1, & -0.4 \leq x \leq -0.2, \\ 1 - |10(x - 0.1)|, & 0 \leq x \leq 0.2, \\ \frac{1}{6} (\mathcal{F}(x, \alpha, a - \delta) + \mathcal{F}(x, \alpha, a + \delta) + 4\mathcal{F}(x, \alpha, a)), & 0.4 \leq x \leq 0.6, \\ 0, & \text{otherwise,} \end{cases}$$

$$\mathcal{G}(x, \beta, z) = e^{-\beta(x-z)^2},$$

$$\mathcal{F}(x, \alpha, a) = \sqrt{\max(1 - \alpha^2(x - a)^2, 0)}.$$

The above constants are set as $a = 0.5$, $z = -0.7$, $\delta = 0.005$, $\alpha = 10$, and $\beta = \log 2/36\delta^2$. The solution includes a smooth but narrow combination of Gaussians, a square, a triangle, as well as an half ellipse.

We compute this example until $t = 8$ on a mesh with 200 cells and show the numerical results by the third order as well as the fifth-order methods in Fig. 1. Both the ADER-DG methods perform well for the four types of waves in the initial data. The numerical solution by the fifth-order method is clearly sharper than that by the third-order method.

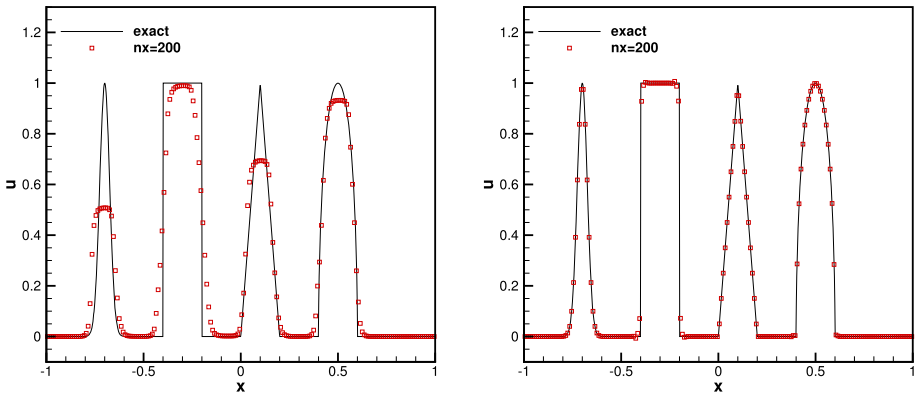


Fig. 1 Linear advection equation by third-order (left) and fifth-order (right) ADER-DG methods. Solutions at $t = 8$

3.1.2 Burgers' equation

Then, we conduct the ADER-DG method for non-linear Burgers' equation (4) coupled with two different initial conditions:

$$u(x, 0) = 0.5 + \sin(\pi x), \quad x \in [0, 2] \quad \text{and} \quad u(x, 0) = \begin{cases} -0.5 & x \leq 0.5, \\ 1 & x \leq 1, \quad x \in [0, 1.5]. \\ 0 & \text{else,} \end{cases}$$

The first case develops a shock; the second one produces a shock as well as a rarefaction at the same time. The numerical solutions of the two examples in comparison with the exact solutions are shown in Fig. 2. The numerical solutions obviously agree well with the exact solutions, and the discontinuities are all well resolved.

3.2 The system case

3.2.1 Testing the order of accuracy

On the basis of the Euler equations (6), we testify the order of accuracy using an example with exact solutions from (Qiu and Shu 2003)

$$\rho(x, t) = 1 + 0.2 \sin(\pi(x - t)), \quad u(x, t) = 1, \quad p(x, t) = 1, \quad x \in [0, 2],$$

and we exert periodic boundary conditions at both ends of the spatial domain. For this example, we only apply the polynomial of degree four. The errors at time $t = 2$ and orders of accuracy are shown in Table 2. The solutions clearly converge at the optimal rate.

3.2.2 Sod problem

The initial conditions are as in (Sod 1978)

$$(\rho, u, p)(x, 0) = \begin{cases} (1, 0, 1) & \text{if } x \leq 0, \\ (0.125, 0, 0.1) & \text{if } x > 0, \end{cases}$$

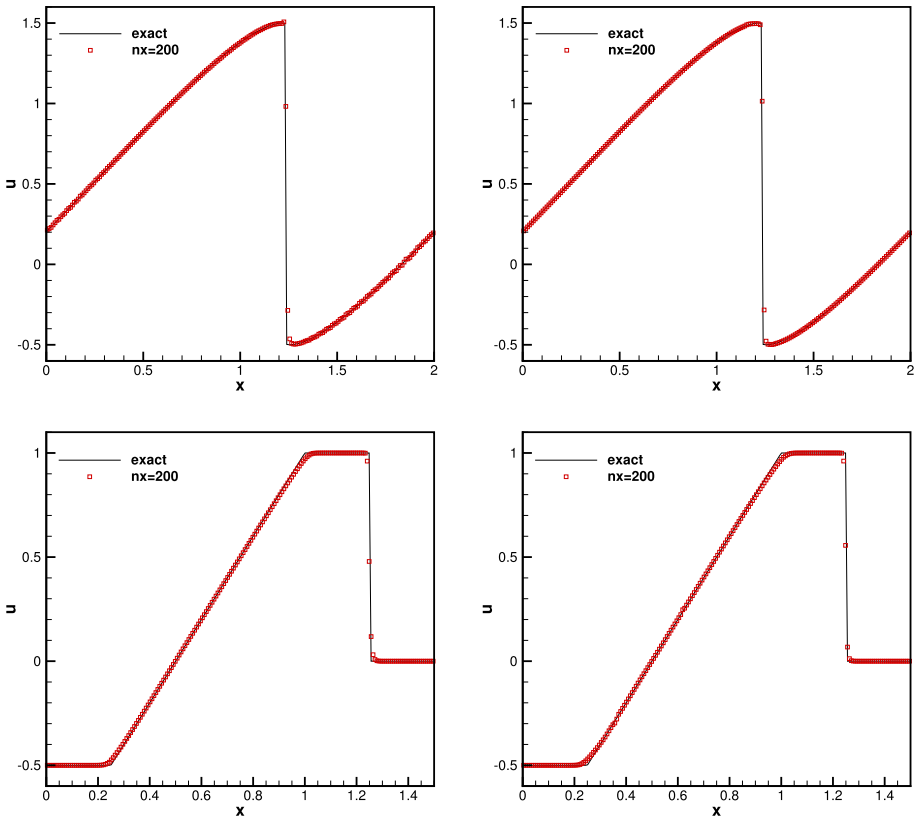


Fig. 2 Burgers’ equation by third-order (left) and fifth-order (right) ADER-DG methods. Top: solutions of the first example at $t = 1.5/\pi$; Bottom: solutions of the second example at $t = 0.5$ (right)

Table 2 The errors and orders of accuracy for density

Cells	L^∞ error	Order	L^1 error	Order	L^2 error	Order
25	7.1557E-09		5.4250E-09		4.6434E-09	
50	1.4620E-09	2.29	5.0198E-10	3.43	5.4977E-10	3.08
100	3.2688E-11	5.48	1.1626E-11	5.43	1.2667E-11	5.44
200	1.2699E-12	4.69	5.8475E-13	4.31	5.2894E-13	4.58

on $[-5, 5]$. This example develops a rarefaction fan, a contact discontinuity as well as a shock at the same time. The computed density at time $t = 2$ is shown in Fig. 3. The numerical results keep sharp discontinuity transition and the contact discontinuities are also considerably well resolved.

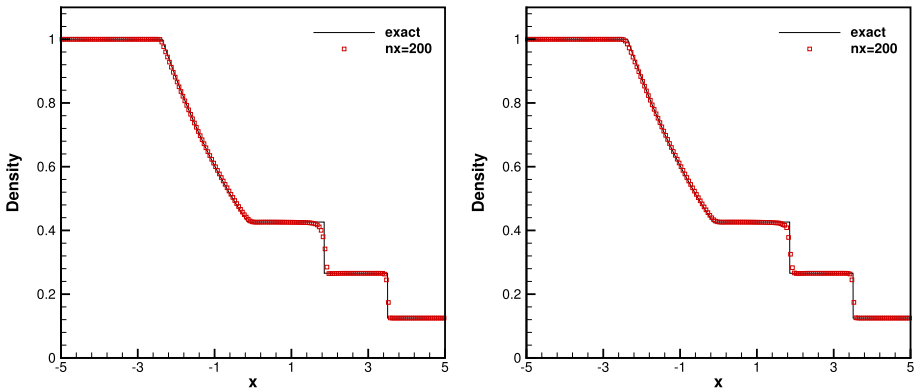


Fig. 3 Sod problem by third-order (left) and fifth-order (right) ADER-DG methods. Density at $t = 2$

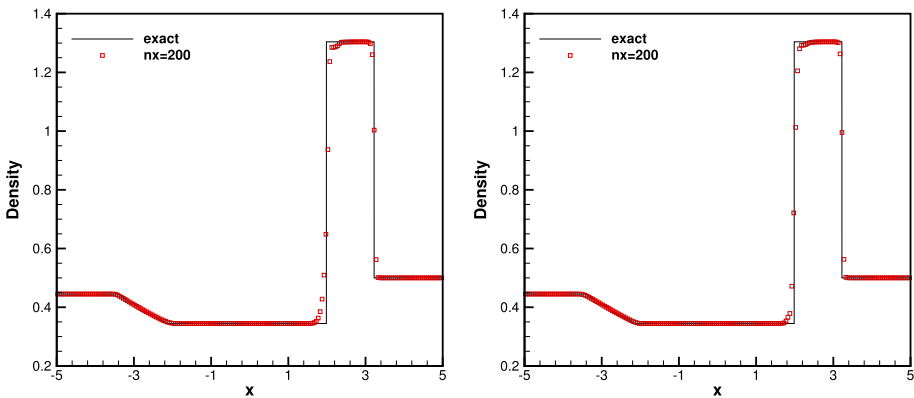


Fig. 4 Lax problem by third-order (left) and fifth-order (right) ADER-DG methods. Density at $t = 1.3$

3.2.3 Lax problem

The initial conditions are defined by

$$(\rho, u, p)(x, 0) = \begin{cases} (0.445, 0.698, 3.528) & \text{if } x \leq 0, \\ (0.5, 0, 0.571) & \text{if } x > 0, \end{cases}$$

on $[-5, 5]$. This example develops a shock as well as a contact discontinuity, which is difficult to resolve accurately. The computed density at time $t = 1.3$ compared with the exact one are shown in Fig. 4. The strong shock is equally well resolved by means of both methods.

3.2.4 Shu–Osher problem

The initial data are given by (Shu and Osher 1988)

$$(\rho, u, p)(x, 0) = \begin{cases} (3.857143, 2.629369, 10.333333) & \text{if } x < -4, \\ (1 + 0.2 \sin(5x), 0, 1) & \text{if } x \geq -4, \end{cases}$$

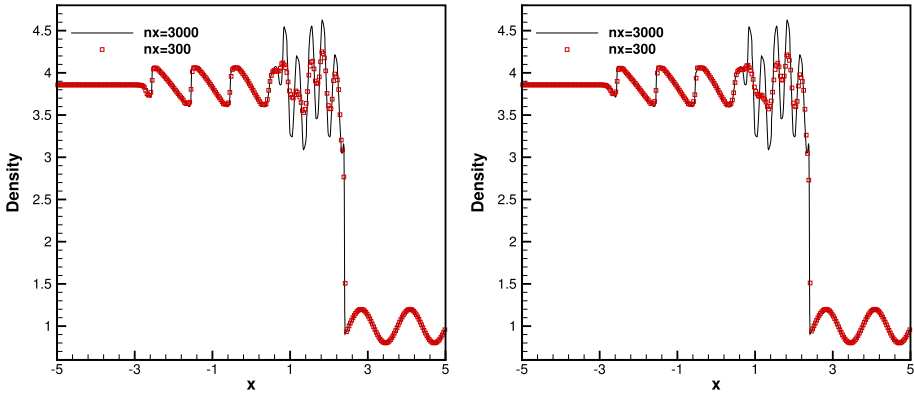


Fig. 5 Shu-Osher problem by third-order (left) and fifth-order (right) ADER-DG methods. Density at $t = 1.8$

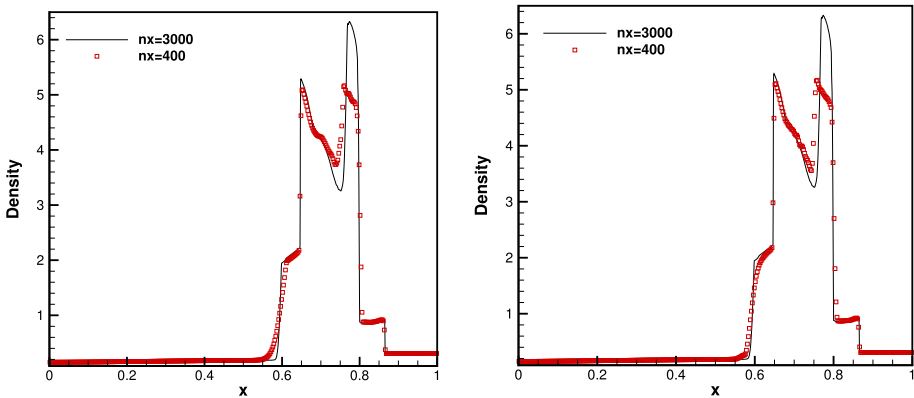


Fig. 6 Blast wave problem by third-order (left) and fifth-order (right) ADER-DG methods. Density at $t = 0.038$

on $[-5, 5]$. This example produces a shock interacting with complex smooth regions. We present the computed density at time $t = 1.8$ in Fig. 5, which fits well with the reference solution.

3.2.5 Blast wave problem

The following example involves interaction of blast waves and its initial conditions are given by (Woodward and Colella 1984)

$$(\rho, u, p)(x, 0) = \begin{cases} (1, 0, 1000) & \text{if } 0 \leq x < 0.1, \\ (1, 0, 0.01) & \text{if } 0.1 \leq x < 0.9, \\ (1, 0, 100) & \text{if } 0.9 \leq x \leq 1, \end{cases}$$

on an unity domain $[0, 1]$. We impose a reflective boundary condition at both ends of the spatial domain, see Woodward and Colella (1984) for more details. We carry out the simulation on a mesh with 400 cells up to time $t = 0.038$ and present the computed density as well as the reference one in Fig. 6. The numerical results obviously keep a steep discontinuity transition.

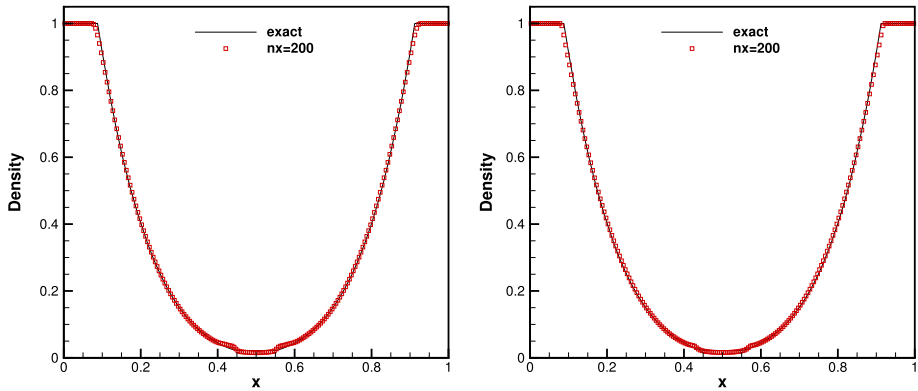


Fig. 7 123 problem by third-order (left) and fifth-order (right) ADER-DG methods. Density at $t = 0.15$

3.2.6 123 problem

The initial conditions are given by (Toro 1999)

$$(\rho, u, p)(x, 0) = \begin{cases} (1, -2, 0.4) & \text{if } x < 0.5, \\ (1, 2, 0.4) & \text{if } x > 0.5, \end{cases}$$

on $[0, 1]$. We present the computed density at time $t = 0.15$ on a mesh with 200 cells in comparison with the exact one in Fig. 7. The central expansion regions are all well resolved by the current methods.

3.2.7 Modified shock/turbulence interaction

Then, we conduct an example (Toro and Titarev 2005) that is actually a modification of the shock/turbulence problem developed in Jiang and Shu (1996b) and Balsara and Shu (2000). The modification mainly includes three parts: (i) a weaker shock wave, (ii) a density fluctuation with frequency four times higher and (iii) a ending time ten times larger. This example produces a right facing shock wave running into a high-frequency density perturbation. As time develops, a shock moves into this density perturbation, which spreads upstream. In practice, this example can also be regarded as an extension of the Shu–Osher problem in Sect. 3.2.4 and is applied to validate a severely oscillatory wave interacting with a shock. Especially, the example here is more appropriate for testing the performance of the current high-order method. We use the following initial conditions

$$(\rho, u, p)(x, 0) = \begin{cases} (1.515695, 0.523346, 1.805) & \text{if } x \leq -4.5, \\ (1 + 0.1 \sin(20\pi x), 0, 1) & \text{if } x > -4.5, \end{cases}$$

on $[-5, 5]$. We implement the computation on a very refined mesh with 1000 cells up to time $t = 5$ and show the computed density in Fig. 8. Both methods produce excellent resolution for these high-frequency oscillations.

In addition, we also compare the CPU time for examples from Sects. 3.2.2–3.2.7 by the third-order ADER-DG method and the third-order RKDG method, see Table 3. As shown in the Table 3, the third-order ADER-DG method can save at least 38% of CPU time, and even save 70% of CPU time compared with the RKDG method with the same order of accuracy.

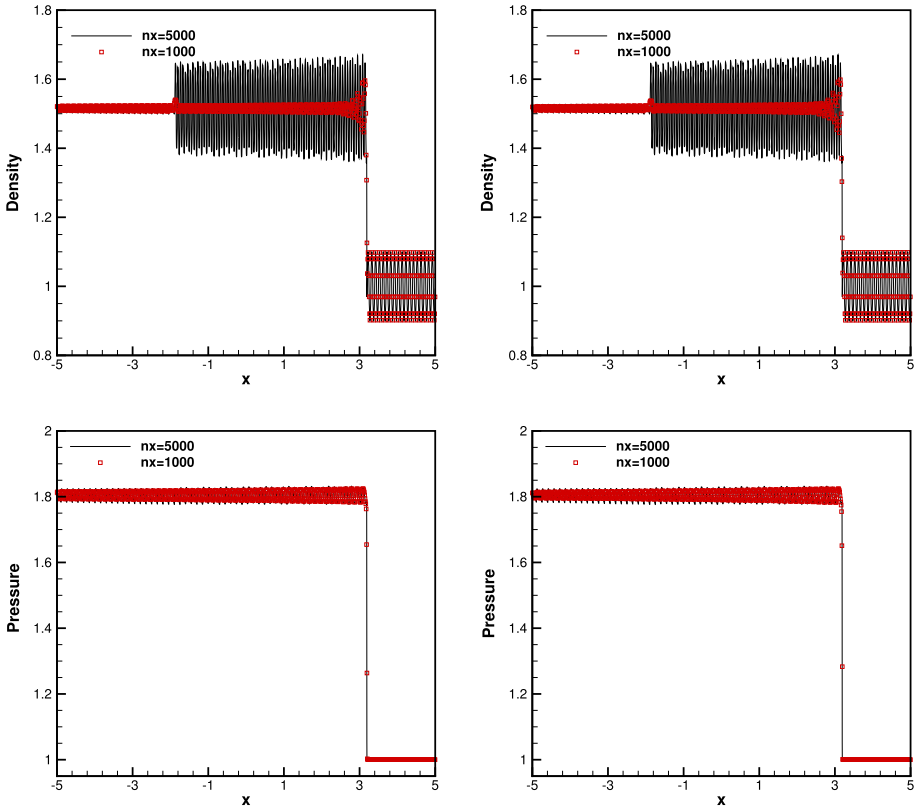


Fig. 8 Modified shock/turbulence interaction problem by third-order (left) and fifth-order (right) ADER-DG methods. Upper: density at $t = 5$; Bottom: pressure at $t = 5$

Table 3 For the CPU time, the gain of ADER-DG in terms of percentage with regard to the RKDG method

Examples	Gain(%)
Sod problem	67.15
Lax problem	69.20
Blast wave problem	66.09
Shu–Osher problem	38.82
123 problem	70.59
Modified shock/turbulence interaction problem	67.63

3.2.8 Vortex evolution problem

In the end, we apply a one-dimensional example similar to the two-dimensional example from Shu (1997). This example here is used to validate the proposed method with regard to the conservation of a vortex for long time evolution. The setup of this example is as follows. The mean flow is $\rho = 1, p = 1, u = 1$ on the spatial domain $[0, 10]$. We add, to the mean flow, an isentropic vortex (perturbation in velocity u and the temperature $T = p/\rho$ and no

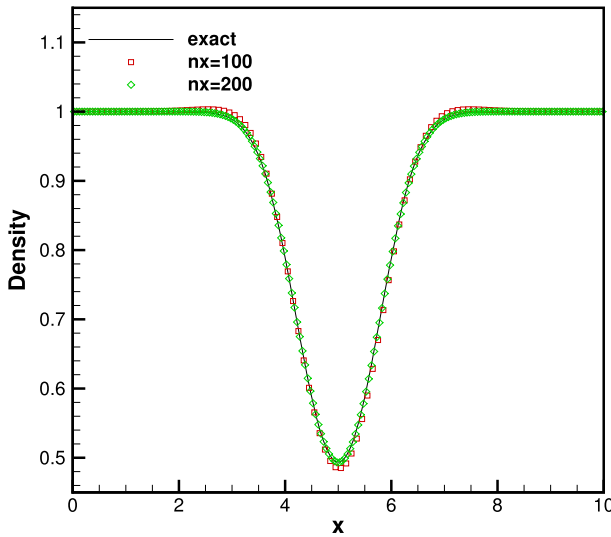


Fig. 9 Vortex evolution problem by third-order ADER-DG method. Density at $t = 100$

perturbation in the entropy $S = p/\rho^\gamma$) as follows

$$\delta u = \frac{\varepsilon}{2\pi} e^{(0.5(1-(x-5)^2))} (x - 5), \quad \delta T = -\frac{(\gamma - 1)\varepsilon^2}{8\gamma\pi} e^{1-(x-5)^2},$$

with $\varepsilon = 5$.

We impose periodic boundary conditions and carry out this example on different meshes with 100 and 200 cells. The numerical results by the third-order ADER-DG method at $t = 100$ (after 10 time periods) are shown in Table 9. As we can see from Table 9, the ADER-DG method can keep a good conservation of the vortex even for a larger time modelling. In addition, with the refinement of the mesh, the conservation of the vortex is improved.

4 Summary and conclusions

In this article, we develop a new DG method with the one-stage ADER approach for the temporal discretization. The key component of this method is to express the spatiotemporal expansion coefficients of the solution through the low-order spatial expansion coefficients using the differential transformation procedure recursively. So, we can obtain the solution in a space–time polynomial form on each space–time control volume starting from the solution at time t^n using the differential transformation procedure repeatedly. Since the numerical solutions are in space–time polynomial forms, we can adopt precise calculations for the numerical fluxes as well as the volume integrations, and avoid using the numerical quadrature rules correspondingly.

The differential transformation procedure is more efficient and the coding is more concise than the Cauchy–Kowalewski procedure. Compared with the RKDG methods, the current method needs less computer memory due to no intermediate stages. Thanks to the explicit one-step nature and the compact stencil, the current ADER-DG method is an ideal candidate for parallel computing on supercomputers. In addition, the proposed method avoids solving

the generalized Riemann problems at inter-cells. Moreover, we can easily proceed to arbitrary high-order accuracy in space and time without much coding effort. In summary, the resulting method is one step, one stage, and fully discrete.

Several classical examples demonstrate the high-order accuracy, good resolution for discontinuous solutions and high computational efficiency. Extension to two-dimensional cases constitutes our ongoing research.

Acknowledgements This research is supported by the National Natural Science Foundation of China (11771228).

Appendix A: The algorithm of the differential transformation procedure for Burgers' equation

Input: $\tilde{u}(k_x, 0)$, $k_x = 0, 1, \dots, k$.
Output: $\tilde{u}(k_x, k_t)$ and $\tilde{f}(k_x, k_t)$, for $k_t = 0, 1, \dots, k$; $k_x = 0, 1, \dots, k - k_t$.
while cell I_j , $j = 1, N$ **do**
 while $k_t = 0, k$ **do**
 while $k_x = 0, k - k_t$ **do**
 $tot_f = 0$
 while $r_t = 0, k_t$ **do**
 while $r_x = 0, k_x$ **do**
 $tot_f = tot_f + \tilde{u}(r_x, r_t) * \tilde{u}(k_x - r_x, k_t - r_t)$
 end while
 end while
 $\tilde{f}(k_x, k_t) = \frac{1}{2} tot_f$ % $f(u) = \frac{1}{2} u^2$
 end while
 if $k_t < k$ **then**
 while $k_x = 0, k - (k_t + 1)$ **do**
 $\tilde{u}(k_x, k_t + 1) = -\frac{k_x + 1}{k_t + 1} * \tilde{f}(k_x + 1, k_t)$
 end while
 end if
 end while
end while

Appendix B: The algorithm of the differential transformation procedure for Euler equations

Input: $\tilde{U}(k_x, 0, 1 : 3)$, $k_x = 0, 1, \dots, k$.
Output: $\tilde{U}(k_x, k_t, 1 : 3)$ and $\tilde{F}(k_x, k_t, 1 : 3)$, for $k_t = 0, 1, \dots, k$; $k_x = 0, 1, \dots, k - k_t$.
while cells I_j , $j = 1, N$ **do**
 $g1 = 0$
 $g1(0, 0) = 1/\tilde{U}(0, 0, 1)$
 while $k_t = 0, k$ **do**
 while $k_x = 0, k - k_t$ **do**
 $tot_g1 = 0$

```

tot_g2 = 0
while r_t = 0, k_t do
  while r_x = 0, k_x do
    tot_g1 = tot_g1 +  $\tilde{U}(r_x, r_t, 1) * g1(k_x - r_x, k_t - r_t)$ 
    tot_g2 = tot_g2 +  $\tilde{U}(r_x, r_t, 2) * \tilde{U}(k_x - r_x, k_t - r_t, 2)$ 
  end while
end while
g2(k_x, k_t) = tot_g2
if k_x + k_t > 0 then
  g1(k_x, k_t) = -tot_g1 /  $\tilde{U}(0, 0, 1)$ 
end if
tot_g3 = 0
tot_g4 = 0
while r_t = 0, k_t do
  while r_x = 0, k_x do
    tot_g3 = tot_g3 +  $g2(r_x, r_t) * g1(k_x - r_x, k_t - r_t)$ 
    tot_g4 = tot_g4 +  $\tilde{U}(r_x, r_t, 2) * g1(k_x - r_x, k_t - r_t)$ 
  end while
end while
g3(k_x, k_t) = tot_g3
g4(k_x, k_t) = tot_g4
g5 = 0
g6 = 0
while r_t = 0, k_t do
  while r_x = 0, k_x do
    g5 = g5 +  $g4(r_x, r_t) * \tilde{U}(k_x - r_x, k_t - r_t, 3)$ 
    g6 = g6 +  $g4(r_x, r_t) * g3(k_x - r_x, k_t - r_t)$ 
  end while
end while
 $\tilde{F}(k_x, k_t, 1) = \tilde{U}(k_x, k_t, 2)$ 
 $\tilde{F}(k_x, k_t, 2) = (\gamma - 1) * \tilde{U}(k_x, k_t, 3) + \left(1 - \frac{\gamma-1}{2}\right) * g3(k_x, k_t)$ 
 $\tilde{F}(k_x, k_t, 3) = \gamma * g5 - \frac{1}{2} * (\gamma - 1) * g6$ 
end while
if k_t < k then
  while k_x = 0, k - (k_t + 1) do
     $\tilde{U}(k_x, k_t + 1, 1) = -\frac{k_x + 1}{k_t + 1} * \tilde{F}(k_x + 1, k_t, 1)$ 
     $\tilde{U}(k_x, k_t + 1, 2) = -\frac{k_x + 1}{k_t + 1} * \tilde{F}(k_x + 1, k_t, 2)$ 
     $\tilde{U}(k_x, k_t + 1, 3) = -\frac{k_x + 1}{k_t + 1} * \tilde{F}(k_x + 1, k_t, 3)$ 
  end while
end if
end while
end while

```

References

- Ayaz F (2003) On the two-dimensional differential transform method. *Appl Math Comput* 143:361–374
- Ayaz F (2004) Solutions of the system of differential equations by differential transform method. *Appl Math Comput* 147:547–567
- Balsara DS, Shu C-W (2000) Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy. *J Comput Phys* 160:405–452
- Cockburn B, Shu C-W (1989) TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. *Math Comput* 52:411–435
- Cockburn B, Shu C-W (1998) The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. *J Comput Phys* 141:199–224
- Cockburn B, Lin S-Y, Shu C-W (1989) TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems. *J Comput Phys* 84:90–113
- Cockburn B, Hou S, Shu C-W (1990) The Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case. *Math Comput* 54:545–581
- Cockburn B, Karniadakis G, Shu C-W (2000) The development of discontinuous Galerkin methods. In: Cockburn B, Karniadakis G, Shu C-W (eds) *Discontinuous Galerkin methods: theory, computation and applications*. Lecture notes in computational science and engineering, part I: overview, vol 11. Springer, Berlin, pp 3–50
- Duan J, Tang H (2020) An efficient ADER discontinuous Galerkin scheme for directly solving Hamilton–Jacobi equation. *J Comput Math* 38(1):58–83
- Dumbser M (2005) Arbitrary high order schemes for the solution of hyperbolic conservation laws in complex domains. Shaker Verlag, Aachen
- Dumbser M, Munz CD (2005a) ADER discontinuous Galerkin schemes for aeroacoustics. *Comput Rendus Mecanique* 333:683–687
- Dumbser M, Munz CD (2005b) Arbitrary high order discontinuous Galerkin schemes. Numerical methods for hyperbolic and kinetic problems. In: Cordier S, Goudon T, Gutnic M, Sonnendrucker E (eds) *IRMA series in mathematics and theoretical physics*. EMS Publishing House, Zurich, pp 295–333
- Dumbser M, Munz CD (2006) Building blocks for arbitrary high order discontinuous Galerkin schemes. *J Sci Comput* 27:215–230
- Dumbser M, Enaux C, Toro EF (2008) Finite volume schemes of very high order of accuracy for stiff hyperbolic balance laws. *J Comput Phys* 227:3971–4001
- Dumbser M, Hidalgo A, Zanotti O (2014) High order space-time adaptive ADER-WENO finite volume schemes for non-conservative hyperbolic systems. *Comput Methods Appl Mech Eng* 268:359–387
- Fambri F, Dumbser M, Zanotti O (2017) Space-time adaptive ADER-DG schemes for dissipative flows: compressible Navier–Stokes and resistive MHD equations. *Comput Phys Commun* 220:297–318
- Fambri F, Dumbser M, Köppel S, Rezzolla L, Zanotti O (2018) ADER discontinuous Galerkin schemes for general-relativistic ideal magnetohydrodynamics. *Mon Not R Astron Soc* 477:4543–4564
- Gottlieb S, Ketcheson DI, Shu C-W (2009) High order strong stability preserving time discretizations. *J Sci Comput* 38:251–289
- Harten A (1987) Uniformly high order accurate essentially non-oscillatory schemes III. *J Comput Phys* 71:231–303
- Jiang GS, Shu C-W (1996a) Efficient implementation of weighted ENO schemes. *J Comput Phys* 126:202–228
- Jiang GS, Shu C-W (1996b) Efficient implementation of weighted ENO schemes. *J Comput Phys* 126:202–228
- Kurnaza A, Oturanc G, Kiris ME (2005) n -Dimensional differential transformation method for solving PDEs. *Int J Comput Math* 82(3):369–380
- Norman MR, Finkel H (2012) Multi-moment ADER-Taylor methods for systems of conservation laws with source terms in one dimension. *J Comput Phys* 231:6622–6642
- Pedro JC, Banda MK, Sibanda P (2014) On one-dimensional arbitrary high-order WENO schemes for systems of hyperbolic conservation laws. *Comput Appl Math* 33:363–384
- Pukhov GE (1982) Differential transforms and circuit theory. *Int J Circuit Theory Appl* 10:265–276
- Qiu J, Shu C-W (2003) Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method: one-dimensional case. *J Comput Phys* 193:115–135
- Qiu J, Dumbser M, Shu C-W (2005) The discontinuous Galerkin method with Lax–Wendroff type time discretizations. *Comput Methods Appl Mech Eng* 194:4528–4543
- Rannabauer L, Dumbser M, Bader M (2018) ADER-DG with a-posteriori finite-volume limiting to simulate tsunamis in a parallel adaptive mesh refinement framework. *Comput Fluids* 000:1–8
- Shu C-W (1997) Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. NASA/CR-97-206253, ICASE Report No. 97-65

- Shu C-W (2016) High order WENO and DG methods for time-dependent convection-dominated PDEs: a brief survey of several recent developments. *J Comput Phys* 316:598–613
- Shu C-W, Osher S (1988) Efficient implementation of essentially non-oscillatory shock capturing schemes. *J Comput Phys* 77(2):439–471
- Sod G (1978) A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J Comput Phys* 27:1–31
- Titarev VA, Toro EF (2002) ADER: arbitrary high order Godunov approach. *J Sci Comput* 17:609–618
- Titarev VA, Toro EF (2005) ADER schemes for three-dimensional non-linear hyperbolic systems. *J Comput Phys* 204(2):715–736
- Toro EF (1999) Riemann solvers and numerical methods for fluid dynamics: a practical introduction. Springer, Berlin
- Toro EF, Titarev VA (2005) TVD fluxes for the high-order ADER schemes. *J Sci Comput* 24(3):285–309
- Toro EF, Millington RC, Nejad LAM (2001) Towards very high order Godunov schemes. In: Toro EF (ed) Godunov methods. Theory and applications, edited review. Kluwer Academic Publishers, Dordrecht, pp 907–940
- Woodward P, Colella P (1984) The numerical simulation of two-dimensional fluid flow with strong shocks. *J Comput Phys* 54(1):115–173
- Zanotti O, Fambri F, Dumbser M, Hidalgo A (2015) Space-time adaptive ADER discontinuous Galerkin finite element schemes with a posteriori sub-cell finite volume limiting. *Comput Fluids* 118:204–224
- Zhou JK (1986) Differential transformation and its applications for electrical circuits. Huazhong University Press, Wuhan

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.