

Your Interlibrary Loan request has been sent by email in a PDF format.

If this PDF arrives with an incorrect OCLC status, please contact lending located below.

Concerning Copyright Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted materials. Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research". If a user makes a request for, or later uses, a photocopy or reproduction for purpose in excess of "fair use", that user may be liable for copyright infringement. This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Interlibrary Loan Services: We Search the World for You...and Deliver!

Interlibrary Loan Services
The Florida State University
711 West Madison Street
Tallahassee, Florida 32306-1005

Lending the FSU Collection: 850.644.4171
James Elliott- ILL- lend@reserves.lib.fsu.edu

Borrowing for the FSU Community: 850.644.4466
Alicia Brown- ill@reserves.lib.fsu.edu

Odyssey: 128.186.59.120 Ariel: 146.201.65.22
Fax: 850.644.3329

A Space-Time Discontinuous Galerkin Spectral Element Method for Nonlinear Hyperbolic Problems*

Chaoxu Pei[†], Mark Sussman[‡] and M. Yousuff Hussaini[§]

*Department of Mathematics, Florida State University
Tallahassee, FL 32306, USA*

[†]cpei@math.fsu.edu

[‡]sussman@math.fsu.edu

[§]myh@math.fsu.edu

Received 25 May 2017

Revised 21 December 2017

Accepted 10 January 2018

Published 9 May 2018

A space-time discontinuous Galerkin spectral element method is combined with two different approaches for treating problems with discontinuous solutions: (i) adding a space-time dependent artificial viscosity, and (ii) tracking the discontinuity with space-time spectral accuracy. A Picard iteration method is employed to solve nonlinear system of equations derived from the space-time DG spectral element discretization. Spectral accuracy in both space and time is demonstrated for the Burgers' equation with a smooth solution. For tests with discontinuities, the present space-time method enables better accuracy at capturing the shock strength in the element containing shock when higher order polynomials in both space and time are used. The spectral accuracy of the shock speed and location is demonstrated for the solution of the inviscid Burgers' equation obtained by the tracking method.

Keywords: Space-time; discontinuous Galerkin; spectral accuracy; shock capturing; shock tracking; Picard iteration.

1. Introduction

We first present a brief history of numerical methods proposed to solve nonlinear hyperbolic problems. In 1989, Shu and Osher [1989] developed essentially nonoscillatory (ENO) numerical methods for hyperbolic conservation laws. These methods can be implemented up to any prescribed spatial and temporal order of accuracy. Unfortunately, these methods are dissipative regardless of the order. Weighted ENO (WENO) schemes, first developed in 1996 [Jiang and Shu (1996); Balsara and Shu (2000)], are less dissipative than ENO methods. In 1999, Yee *et al.* [1999] proposed a numerical method that is a hybrid of the low-dissipative fourth order method

*Preprint submitted to *International Journal of Computational Methods* December 21, 2017.

[†]Corresponding author.

(fourth-order in space and time) of Gustafsson and Olsson [1995] with a shock capturing method developed by Harten [1978]. Improvements upon the hybrid approach of Yee *et al.* [1999] have since followed. Ren *et al.* [2003] hybridized the compact scheme of Pirozzoli [2002] with the WENO scheme of Jiang and Shu [1996]. Sjögren and Yee [2004] also hybridized a low-dissipative fourth-order method of Gustafsson and Olsson [1995] with the multi-resolution method developed by Harten [1978]. The difference between Sjögren and Yee's [2004] method and the hybrid method of Yee *et al.* [1999] is that the smoothness sensor in Sjögren and Yee's work did not depend on empirical tuning parameters. Instead, sensors were developed by looking at the coefficients of the wavelet transform [Gerritsen and Olsson (1996); Mallat and Zhong (1992); Daubechies (1992); Mallat (1999)]. Reisner *et al.* [2013] developed a sensor that switched on or off an artificial viscosity term depending on the solutions of a linear scalar reaction-diffusion equation, which is referred to as *C*-method. With artificial viscosity deactivated, their method reduced to either a second-order finite element method or a fifth-order WENO method. Abbassi *et al.* [2014] proposed an entropy-based artificial viscosity for spectral collocation method. To smoothen artificial viscosity, an element-level filtering approach is introduced.

Besides methods that switch between spatially dissipative and relatively less dissipative discretization schemes, methods have been developed for limiting the order of the discretization in time. Duraisamy *et al.* [2003] developed time limiters for the second-order two-stage diagonally implicit Runge–Kutta (DIRK2) and the trapezoid time discretization methods. They implemented the fifth-order monotonicity preserving method of Suresh and Huynh [1997] for the spatial discretization. Concurrent with the development of shock capturing methods, there have been the development of methods for tracking discontinuities in hyperbolic conservation laws. We refer to the paper by Touil *et al.* [2007] and the references therein; they resorted to refine the mesh in discontinuous regions by introducing an inordinate number of additional degrees of freedom.

In this work, we develop a space-time discontinuous Galerkin (DG) spectral element method. DG methods in space have been developed for solving a wide range of physical problems, especially for dealing with flows with discontinuities computed on unstructured meshes. Detailed surveys can be found in, for example, Cockburn *et al.* [2000], Canuto *et al.* [2006], Hesthaven and Warburton [2008], Kopriva [2009], and Karniadakis and Sherwin [2013]. Recently, space-time DG finite element methods have been explored by Sudirham *et al.* [2006], Klaij *et al.* [2006], van der Vegt and Sudirham [2008], Sollie *et al.* [2011], and Rhebergen *et al.* [2013], for problems that require moving and deforming meshes. Space-time DG schemes result in implicit time integration and naturally allow for time adaptivity. Lörcher *et al.* [2007] propose an explicit DG scheme for inviscid compressible flow in one space dimension, in which a Taylor expansion in space and time to define a space-time polynomial is used to obtain an arbitrary order of accuracy in space and time. Their approach require a local stability criterion to be satisfied in every grid zone.

Motivated by the properties of space-time DG finite element methods [Sudirham *et al.* (2006); Klaij *et al.* (2006); van der Vegt and Sudirham (2008); Sollie *et al.* (2011); Rhebergen *et al.* (2013)], we present a space-time DG spectral element method for solving nonlinear hyperbolic problems, which is integrated with shock capturing and shock tracking procedures. If the position of a discontinuity is initially unknown, then we employ a sensor proposed by Persson and Peraire [2006] to locate the troublesome element wherein the unresolved steep gradient/shock causes Gibbs phenomenon. Since the proposed method is in the space-time framework, the amount of artificial viscosity is computed at each time “node” independently within each space-time slab, which allows one to take a large time step with optimal diffusion to capture strong shock without Gibbs phenomenon. By increasing the polynomial order in both space and time, one can resolve the shock strength, i.e., the extrema on either side of the shock. If the initial position of a discontinuity is known, or if the solution is smooth everywhere, then the present method is spectrally accurate in both space and time. We note that the present method does not align the computational elements with a discontinuity, which makes it distinguished from a body fitted method [Sudirham *et al.* (2006); van der Vegt and Sudirham (2008); Rhebergen and Cockburn (2012)]. Instead, two global transformations: a backward transformation and a forward transformation, are applied, which combines an Eulerian description with a Lagrangian description. In the proposed tracking method, the discontinuity may cut an element in such a way that one piece can be very tiny. Small cut-cells do not affect the spectral accuracy of the proposed tracking method. In order to deal with nonlinear terms, we employ a Picard iteration procedure [Benzi and Olshanskii (2006); Rhebergen and Cockburn (2012)]. At each iteration, a linear system of equations, derived from the DG spectral element method, is solved.

The organization of this paper is as follows. In Sec. 2, we define the nonlinear initial boundary-value problem to be solved by a space-time DG spectral element method. The space-time discretization framework is introduced in Sec. 3. Two treatments for shocks are then presented in Sec. 4. The outline of the proposed method is given in Sec. 5. Numerical experiments are presented in one spatial dimension in Sec. 6 in order to evaluate the accuracy and efficiency of the present shock treatment methods. The summary of results and conclusions are provided in Sec. 7.

2. Problem Formulation

On a Cartesian domain Ω , we consider a scalar conservation law of the form,

$$u_t + \nabla \cdot \mathbf{F}(u) = 0 \quad (\mathbf{x}, t) \in \Omega \times [0, T], \quad (1)$$

$$u(\mathbf{x}, t) = u_b, \quad (2)$$

$$u(\mathbf{x}, 0) = u_0, \quad (3)$$

where $\mathbf{F}(u)$ is the vector of fluxes in the spatial directions, u_0 is the initial data and u_b denotes the boundary data.

It is well known that such problems may develop discontinuous solutions even if the initial values are smooth. In order to suppress the Gibbs-type oscillations caused by applying higher order numerical methods to discontinuous solutions, one can either introduce a dissipative term [Persson and Peraire (2006)], referred to as the shock capturing method, or track the discontinuities explicitly [Touil *et al.* (2007)] when the initial location of a discontinuity is known.

In order to uniform notations of two procedures for treating shocks, we solve the modified equation of Eq. (1) as follows:

$$u_t + \nabla \cdot \mathbf{F}(u) = \nabla \cdot (\epsilon(u) \nabla u) \quad (\mathbf{x}, t) \in \Omega \times [0, T], \quad (4)$$

where $\epsilon(u)$ is a matrix with nonnegative components depending on the solution u . Note that $\epsilon(u) = 0$ for the shock tracking method in the following.

3. Space-Time Discretization

3.1. Space-time tessellation

In a space-time discretization, we introduce a *space-time domain* \mathcal{E} as $\mathcal{E} = \Omega \times [t^0, T]$. A point in the space-time domain, $\bar{\mathbf{x}} \in \mathcal{E}$, has coordinates (\mathbf{x}, \mathbf{t}) . First, we partition the time interval $[t^0, T]$ uniformly by the time levels $0 = t^0 < t^1 < \dots < t^{E(t)} = T$. The space-time domain \mathcal{E} is then divided into $E^{(t)}$ *space-time slabs*. The n th space-time slab is denoted as $\mathcal{E}^n = \mathcal{E} \cap I^n$, where $I^n = [t^n, t^{n+1}]$ is the n th time interval with length $\Delta t = t^{n+1} - t^n$. We then divide the spatial domain Ω into $E^{(x)} \times E^{(y)}$ nonoverlapping spatial elements. Let Ω_e^n and Ω_e^{n+1} be the spatial element e at time level t^n and t^{n+1} , respectively. A space-time element \mathcal{K}_e^n is then obtained by connecting Ω_e^n and Ω_e^{n+1} . The tessellation of the space-time domain is denoted as \bar{T}_h .

In each space-time slab \mathcal{E}^n , the time interval $I^n = [t^n, t^{n+1}]$ is divided into $p^{(t)}$ subintervals by choosing Legendre Gauss-Lobatto points t_m^n for $m = 0, \dots, p^{(t)}$, with $t^n = t_0^n < t_1^n < \dots < t_{p^{(t)}}^n = t^{n+1}$. In the following, the length $\Delta t = t^{n+1} - t^n$ is referred to as a *time step size* while $\Delta t_m = t_{m+1}^n - t_m^n$ if referred to as a *time substep size*. Figure 1 is an illustration of space-time discretization.

Remark. The definitions of the *space-time domain*, and *space-time slabs* are following the notations presented by Sudirham and his collaborators [Sudirham *et al.* (2006); van der Vegt and Sudirham (2008)]. The *space-time element* defined in [Sudirham *et al.* (2006); van der Vegt and Sudirham (2008)] is constructed by moving and deforming spatial elements, while the one introduced in this work is constructed by stationary spatial elements. By choosing fixed spatial elements, the present space-time method is flexible to implement the spectral element method in the space-time domain, which will achieve spectral accuracy in both space and time in the region where the solution is smooth. This makes the present work

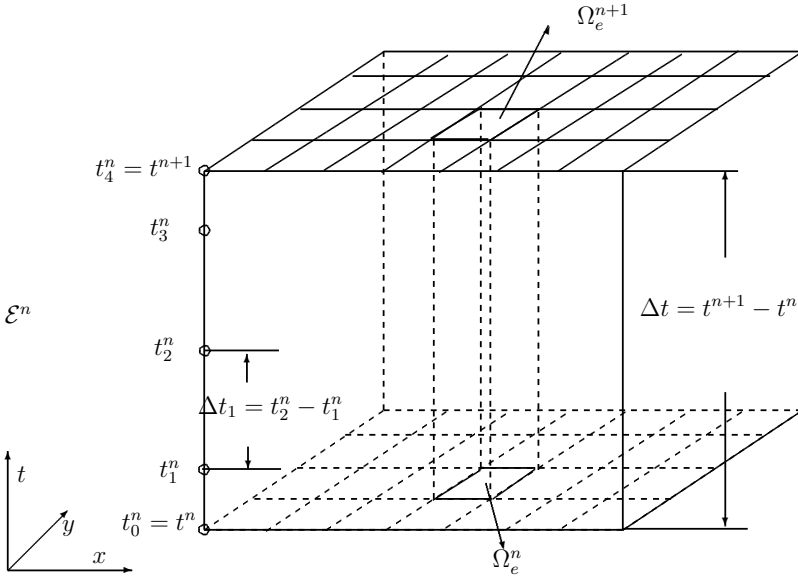


Fig. 1. Example of a space-time slab \mathcal{E}^n . The time interval $[t^n, t^{n+1}]$ is divided into four subintervals by choosing Legendre Gauss-Lobatto points t_m^n for $m = 0, \dots, 4$. $\Delta t = t^{n+1} - t^n$ is referred to as a *time step size* while $\Delta t_m = t_{m+1}^n - t_m^n$ is referred to as a *time substep size*. The small rectangle cube is denoted the space-time element \mathcal{K}_e^n by connecting Ω_e^n and Ω_e^{n+1} .

distinguished from the space-time schemes proposed by Sudirham and his collaborators [Sudirham *et al.* (2006); van der Vegt and Sudirham (2008)].

3.2. Function spaces and notation

First, we introduce a transformation, $\bar{\mathbf{x}} = \mathbf{X}(\xi, \eta)$, which defines a mapping that connects a space-time element $\mathcal{K} \in \mathbb{R}^2$ ($\bar{\mathbf{x}} = (x, t) \in \mathcal{K}$) and the reference square $\mathcal{R} = [-1, 1] \times [-1, 1]$ ($(\xi, \eta) \in \mathcal{R}$). In a space-time domain $\mathcal{E} \in \mathbb{R}^2$, each space-time element is a square. Thus, the transformation $(x, t) = \mathbf{X}(\xi, \eta)$ [Kopriva (2009)] is linear in each coordinate direction and is written as

$$\mathbf{X}(\xi, \eta) = \frac{1}{4} \{ \mathbf{x}^1(1 - \xi)(1 - \eta) + \mathbf{x}^2(1 + \xi)(1 - \eta) + \mathbf{x}^3(1 + \xi)(1 + \eta) + \mathbf{x}^4(1 - \xi)(1 + \eta) \}, \quad (5)$$

where $\{\mathbf{x}^j\}_{j=1, \dots, 4}$ are four corners of a space-time element \mathcal{K} , which are numbered counter clockwise.

Let us introduce two function spaces, V_h and Σ_h , associated with the tessellation $\bar{\mathcal{T}}_h$,

$$V_h = \{ \nu \in L^2(\mathcal{E}) : \nu|_{\mathcal{K}} \circ \mathbf{X}_{\mathcal{K}} \in \mathcal{P}^p(\mathcal{R}), \forall \mathcal{K} \in \bar{\mathcal{T}}_h \}, \quad (6)$$

$$\Sigma_h = \{ \boldsymbol{\tau} \in (L^2(\mathcal{E}))^{1+1} : \boldsymbol{\tau}|_{\mathcal{K}} \circ \mathbf{X}_{\mathcal{K}} \in (\mathcal{P}^p(\mathcal{R}))^{1+1}, \forall \mathcal{K} \in \bar{\mathcal{T}}_h \}, \quad (7)$$

where $\mathcal{P}^p(\mathcal{R})$ is the set of all polynomials of degree at most $p = (p^{(x)}, p^{(t)})$ on \mathcal{R} , with $p^{(x)}$ in the spatial direction and $p^{(t)}$ in the time direction.

Spectral element methods with nonperiodic boundaries use orthogonal polynomial approximations of the solution [Canuto *et al.* (2006); Kopriva (2009)]. In the space-time domain $\mathcal{E} \in \mathbb{R}^2$, we introduce a tensor product basis, $\ell_i(\xi)\ell_j(\eta) \in \mathcal{P}^p(\mathcal{R})$, such that

$$\ell_i(\xi) = \prod_{\substack{k=0 \\ k \neq i}}^{p^{(x)}} \frac{\xi - \xi_k}{\xi_i - \xi_k}, \quad \ell_j(\eta) = \prod_{\substack{k=0 \\ k \neq j}}^{p^{(t)}} \frac{\eta - \eta_k}{\eta_j - \eta_k}, \quad (8)$$

where $\{\xi_i\}_{i=0,\dots,p^{(x)}}$ and $\{\eta_j\}_{j=0,\dots,p^{(t)}}$ are the sets of Legendre Gauss–Lobatto points.

Due to the discontinuous approximation spaces, the approximations are double valued across element boundaries. We introduce the average $\{\{\cdot\}\}$ and jump $[\![\cdot]\!]$ operators. Considering two adjacent elements \mathcal{K}_j^n and \mathcal{K}_{j+1}^n , such that $\mathcal{Q}_{j,j+1}^n = \mathcal{K}_j^n \cap \mathcal{K}_{j+1}^n$. Let $\bar{\mathbf{n}}_{\mathcal{K}_j^n}$ and $\bar{\mathbf{n}}_{\mathcal{K}_{j+1}^n}$ denote the corresponding outward unit normal of \mathcal{K}_j^n and \mathcal{K}_{j+1}^n on the inter-element face $\mathcal{Q}_{j,j+1}^n$. Then the average $\{\{\cdot\}\}$ and jump $[\![\cdot]\!]$ operators are defined,

$$\{\{\nu\}\} = (\nu_{\mathcal{K}_j^n}^+ + \nu_{\mathcal{K}_{j+1}^n}^-)/2, \quad \{\{\tau\}\} = (\tau_{\mathcal{K}_j^n}^+ + \tau_{\mathcal{K}_{j+1}^n}^-)/2, \quad (9)$$

$$[\![\nu]\!] = \nu_{\mathcal{K}_j^n}^+ \bar{\mathbf{n}}_{\mathcal{K}_j^n} + \nu_{\mathcal{K}_{j+1}^n}^- \bar{\mathbf{n}}_{\mathcal{K}_{j+1}^n}, \quad [\![\tau]\!] = \tau_{\mathcal{K}_j^n}^+ \cdot \bar{\mathbf{n}}_{\mathcal{K}_j^n} + \tau_{\mathcal{K}_{j+1}^n}^- \cdot \bar{\mathbf{n}}_{\mathcal{K}_{j+1}^n}, \quad (10)$$

where $\nu_{\mathcal{K}_j^n}^+$ denotes the left value at the inter-element face, and $\nu_{\mathcal{K}_{j+1}^n}^-$ denotes the right value at the inter-element face.

3.3. Space-time formulation

In the space-time domain $\mathcal{E} \in \mathcal{R}^2$, we introduce a vector function \mathbf{B} and a matrix \mathbf{A} , as follows:

$$\mathbf{B} = (\partial F(u)/\partial u, 1), \quad \mathbf{A} = \begin{pmatrix} \epsilon(u) & 0 \\ 0 & 0 \end{pmatrix}. \quad (11)$$

We then transform (4) into a space-time formulation as follows:

$$\nabla \cdot (\mathbf{B}u - \mathbf{A}\nabla u) = 0, \quad \mathbf{x} \in \mathcal{E}, \quad (12)$$

where ∇ denotes the gradient operator with respect to $\bar{\mathbf{x}}$ in the space-time domain, whereas ∇ is the gradient operator with respect to the spatial variables.

To solve Eq. (12), we apply a Picard iteration scheme [Benzi and Olshanskii (2006); Rhebergen and Cockburn (2012)] for which at each Picard iteration a linear advection-diffusion equation has to be solved in the space-time framework. That is, at k th Picard iteration, we solve,

$$\nabla \cdot (\mathbf{B}^{(k)}u^{(k+1)} - \mathbf{A}^{(k)}\nabla u^{(k+1)}) = 0. \quad (13)$$

4. Procedures for Treating Shocks

4.1. Shock capturing

Since the idea of adding artificial viscosity is to eliminate Gibbs oscillations, it is crucial to determine where the solution is discontinuous and how much artificial viscosity is needed. The smoothness indicator introduced by Persson and Peraire [2006], which is similar to error indicators for adaptation in spectral element methods [Mavriplis (1994)], assumes that the polynomial expansion has a similar behavior to the Fourier expansion. The smoothness indicator introduced by Persson and Peraire [2006] is defined as follows:

$$S_K = \frac{\int_K (u_h - \hat{u}_h)^2 dK}{\int_K (u_h)^2 dK}, \quad (14)$$

where u_h is the approximation of order $p^{(\mathbf{x})}$ and \hat{u}_h is the truncated representation of u_h at order $p^{(\mathbf{x})} - 1$. If a function is continuous but its derivative is not continuous, then the coefficients in a Fourier expansion decay like $1/(p^{(\mathbf{x})})^2$. Since the sensor (14) involves square quantities, the approximation u_h is at most C^0 for element K if $S_K > 1/(p^{(\mathbf{x})})^4$.

Once a shock is detected in a spatial element, we determine the amount of artificial viscosity by the following elementwise-constant formulation [Persson and Peraire (2006)]:

$$\epsilon_K = \begin{cases} 0, & s_K < s_0 - \kappa, \\ \frac{\epsilon_0}{2} \left(1 + \sin \frac{\pi(s_K - s_0)}{2\kappa} \right), & s_0 - \kappa \leq s_K \leq s_0 + \kappa, \\ \epsilon_0, & s_K > s_0 + \kappa, \end{cases} \quad (15)$$

where $s_K = \log_{10} S_K$, ϵ_0 is a maximum value, s_0 and κ are empirical tuning parameters. The parameter s_0 should be scaled as $s_0 \sim -4 \log_{10} p^{(\mathbf{x})}$ for the coefficients of the polynomial expansion decay like $1/(p^{(\mathbf{x})})^4$.

The present space-time discontinuous Galerkin scheme is a fully implicit method in each space-time slab. Thus, we introduce a spatio-temporal artificial diffusion which employ the smoothness sensor (14) in each time level, t_m^n with $m = 0, \dots, p^{(t)}$, in the space-time slab \mathcal{E}^n . The amount of artificial viscosity at each time level is computed by Eq. (15). However, the amount of artificial viscosity is recomputed in each Picard iteration step since the solution is updated in each Picard iteration.

Remark.

- We further remark on the distinction between our space-time shock capturing method and ENO/WENO methods. Our space-time method has the property that for a sufficiently resolved mesh, and in space-time elements where derivatives (up to any order) of the solution are all continuous, then the local truncation error of our method is spectrally accurate and the method has zero artificial dissipation. In contrast, for shock capturing methods such as ENO [Kadioglu (2016)], WENO,

MUSCL, or PPM [Kadioglu (2011)] (see also [Layton and Minion (2004)]), artificial dissipation will be added at extrema (either extrema in the solution, or extrema in the high derivatives of the solution). In other words, for methods such as ENO or WENO, even if the underlying solution has all continuous derivatives but has extrema, then numerical dissipation will be implicitly added.

- Another distinction between our approach for shock capturing and the strategy taken by ENO/WENO methods is that our space-time discretization for the Hyperbolic terms remains unchanged so that interpolation coefficients can be pre-computed for efficiency whereas the stencil for ENO/WENO methods is adaptive resulting in a slower method; we refer the reader to Persson and Peraire [2006] for a further discussion on this manner.
- The extension of the shock capturing method to multiple dimensions is straightforward. However, an alternative solver that is faster than our present solver (a combination of Picard iteration and direct matrix gaussian elimination algorithm (see (31))) should be developed.

4.2. Shock tracking

Discontinuous initial data produces Gibbs-type oscillations, which persist at least for a short time even in the artificial viscosity method. To avoid this spurious behavior, one can either smooth the discontinuous initial profiles, e.g., a hyperbolic-tangent smoothing [Reisner *et al.* (2013)], or track the discontinuity explicitly [Touil *et al.* (2007)].

Incorporating the shock tracking procedure in the present space-time discontinuous Galerkin spectral element method follows the work of Pei *et al.* [2017]. The idea is to combine a Eulerian description with a Lagrangian description by two global transformations, a backward transformation and a forward transformation. For one spatial dimension problems, two global transformations are defined as follows:

- the backward transformation,

$$\tilde{x}(x, t) = x - \int_{t_n}^t W(s) ds, \quad (16)$$

- the forward transformation,

$$x(\tilde{x}, t) = \tilde{x} + \int_{t_n}^t W(s) ds. \quad (17)$$

there, x is the spatial variable corresponding to the fixed frame of reference and \tilde{x} is the coordinate of the moving frame of reference. W is the normal interface velocity, i.e., the shock speed. In the moving frame of reference, the transformed governing equation (4) is then obtained,

$$\tilde{u}_t + \tilde{\nabla} \cdot (F(\tilde{u}) - W\tilde{u}) = \tilde{\nabla} \cdot (\epsilon(\tilde{u})\tilde{\nabla}\tilde{u}). \quad (18)$$

The corresponding space-time formulation is written as

$$\tilde{\nabla} \cdot (\tilde{\mathbf{B}}^* \tilde{u} - \tilde{\mathbf{A}} \tilde{\nabla} \tilde{u}) = 0, \quad (19)$$

where $\tilde{\nabla}$ is the gradient operator with respect to $\tilde{\mathbf{x}} = (\tilde{x}, t)$, and $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}^*$ are defined as follows:

$$\tilde{\mathbf{B}}^* = (\partial F(\tilde{u})/\partial \tilde{u} - W, 1), \quad \tilde{\mathbf{A}} = \begin{pmatrix} \epsilon(\tilde{u}) & \epsilon(\tilde{u}) & 0 \\ 0 & 0 & 0 \end{pmatrix}. \quad (20)$$

By applying a Picard iteration scheme [Benzi and Olshanskii (2006); Rhebergen and Cockburn (2012)], a linear advection-diffusion equation has to be solved in the space-time framework. At k th Picard iteration, we solve,

$$\tilde{\nabla} \cdot (\tilde{\mathbf{B}}^{*,(k)} \tilde{u}^{(k+1)} - \tilde{\mathbf{A}}^{(k)} \tilde{\nabla} \tilde{u}^{(k+1)}) = 0. \quad (21)$$

Remark.

- The difference between the present shock tracking method and the work of Pei *et al.* [2017] is the vector $\tilde{\mathbf{B}}^*$, where the strong nonlinearity $F(\tilde{u})/\partial \tilde{u}$ causes the collision of characteristics by a tiny cut-cell generations. This problem is manifested in the Picard iteration not converging unless the time step is sufficiently refined. However, the present space-time method is fully implicit and naturally allow for time adaptivity.
- The computed shock speed, W , needs to be updated along with the solution in each space-time slab due to its dependence on the solution on either side of the front and its influence on the transformations.
- The interface is updated by a level set procedure which captures the location of the discontinuity [Sussman *et al.* (1994); Sussman and Hussaini (2003); Touil *et al.* (2007)].
- In multiple dimensions, it is possible that the transformations become singular. It would then be necessary to reduce the spatial and temporal order. We refer the reader to Pei *et al.* [2017] for more information.

4.3. Discretization of DG spectral element method

To derive the space-time DG weak formulation for either Eq. (13) or Eq. (21), we follow the same approach presented by [Cockburn and Shu (1998)]. In the following, we take Eq. (13) as an example to describe the space-time DG weak formulation. By introducing an auxiliary variable σ , we rewrite Eq. (13) into a first-order system

$$\sigma^{(k+1)} = \nabla u^{(k+1)}, \quad (22)$$

$$\nabla \cdot (\mathbf{B}^{(k)} u^{(k+1)} - \mathbf{A}^{(k)} \sigma^{(k+1)}) = 0. \quad (23)$$

We then present the derivation of the space-time DG spectral element method for Eqs. (22) and (23). Without loss of generality, we drop the iteration symbols, i.e., the superscript $^{(k)}$ and $^{(k+1)}$.

4.3.1. Space-time DG weak formulations

To derive the space-time DG weak formulation for the auxiliary equation, we multiply (22) with a test function $\boldsymbol{\tau} \in \boldsymbol{\Sigma}_h$, substitute u with the approximation $u_h \in V_h$, and integrate over a space-time element $\mathcal{K}_e^n \in \mathcal{T}_h^n$. Next, we perform integration by parts once to obtain,

$$\int_{\mathcal{K}_e^n} \boldsymbol{\sigma}_h \cdot \boldsymbol{\tau} d\mathcal{K}_e^n = - \int_{\mathcal{K}_e^n} u_h \nabla \cdot \boldsymbol{\tau} d\mathcal{K}_e^n + \int_{\partial\mathcal{K}_e^n} \hat{u}_h^\epsilon \bar{\mathbf{n}} \cdot \boldsymbol{\tau} d\partial\mathcal{K}_e^n, \quad (24)$$

where $\bar{\mathbf{n}}$ denotes the outward unit normal vector of the space-time element \mathcal{K}_e^n and \hat{u}_h^ϵ denotes numerical flux.

Similarly, we derive the space-time DG weak formulation for (23) by multiplying it with a test function $\nu \in V_h$, substituting u and $\boldsymbol{\sigma}$ with the approximations $u_h \in V_h$ and $\boldsymbol{\sigma}_h \in \boldsymbol{\Sigma}_h$, and integrating over a space-time element $\mathcal{K}_e^n \in \mathcal{T}_h^n$. After performing integration by parts once, we have,

$$- \int_{\mathcal{K}_e^n} (\mathbf{B}u_h - \mathbf{A}\boldsymbol{\sigma}_h) \cdot \nabla \nu d\mathcal{K}_e^n + \int_{\partial\mathcal{K}_e^n} (\mathbf{B}\hat{u}_h - \mathbf{A}\hat{\boldsymbol{\sigma}}_h^\epsilon) \cdot \bar{\mathbf{n}} \nu d\partial\mathcal{K}_e^n = 0, \quad (25)$$

where \hat{u}_h and $\hat{\boldsymbol{\sigma}}_h^\epsilon$ are numerical fluxes.

4.3.2. Definition of numerical fluxes

We separate the numerical fluxes into two categories: the convective fluxes \hat{u}_h ; the diffusive fluxes related to the artificial viscosity term, \hat{u}_h^ϵ and $\hat{\boldsymbol{\sigma}}_h^\epsilon$.

For the convective fluxes, we choose the upwind flux [Sudirham *et al.* (2006)],

$$\mathbf{B}\hat{u}_h = \{\{\mathbf{B}\}\}\{\{u_h\}\} + \frac{1}{2}|\{\{\mathbf{B}\}\} \cdot \bar{\mathbf{n}}|\llbracket u_h \rrbracket. \quad (26)$$

For the diffusive fluxes \hat{u}_h^ϵ and $\hat{\boldsymbol{\sigma}}_h^\epsilon$, there are several different formulations presented Arnold *et al.* [2001]. Here, we choose the local discontinuous Galerkin (LDG) fluxes [Cockburn and Shu (1998)],

$$\hat{u}_h^\epsilon = \{\{u_h\}\} - \boldsymbol{\beta} \cdot \llbracket u_h \rrbracket, \quad (27)$$

$$\hat{\boldsymbol{\sigma}}_h^\epsilon = \{\{\boldsymbol{\sigma}_h\}\} + \boldsymbol{\beta} \cdot \llbracket \boldsymbol{\sigma}_h \rrbracket - \alpha_j \llbracket u_h \rrbracket, \quad (28)$$

where $\boldsymbol{\beta} \cdot \mathbf{n} = \text{sign}(\mathbf{v} \cdot \mathbf{n})/2$ ([Cockburn *et al.* (2001)]) with $\mathbf{v} = (1, 1, 1)^\top$ and $\alpha_j \geq 0$ is referred as the stabilization parameter.

4.3.3. The space-time slab linear system

From (24) and (25), we note that the outward normal vectors $\bar{\mathbf{n}}$ at the faces of K^n and $K^{(n-1)}$ in a space-time slab \mathcal{E}^n are $(0, 1)$ and $(0, -1)$, respectively. This makes the space-time slab \mathcal{E}^n only depend on its previous space-time slab \mathcal{E}^{n-1} , which leads to weakly coupled space-time slabs. We then perform summation over all elements of the tessellation \mathcal{T}_h^n in one space-time slab \mathcal{E}^n for (24)

and (25),

$$\sum_e \int_{\mathcal{K}_e^n} \boldsymbol{\sigma}_h \cdot \boldsymbol{\tau} d\mathcal{K}_e^n = \sum_e \left(- \int_{\mathcal{K}_e^n} u_h \boldsymbol{\nabla} \cdot \boldsymbol{\tau} d\mathcal{K}_e^n + \int_{\partial \mathcal{K}_e^n} \hat{u}_h^\epsilon \bar{\mathbf{n}} \cdot \boldsymbol{\tau} d\partial \mathcal{K}_e^n \right), \quad (29)$$

$$\sum_e \left(- \int_{\mathcal{K}_e^n} (B u_h - A \boldsymbol{\sigma}_h) \cdot \boldsymbol{\nabla} \nu d\mathcal{K}_e^n + \int_{\partial \mathcal{K}_e^n} (B \hat{u}_h - A \hat{\boldsymbol{\sigma}}_h^\epsilon) \cdot \bar{\mathbf{n}} \nu d\partial \mathcal{K}_e^n \right) = 0. \quad (30)$$

The weak coupling between space-time slabs reduces the number of unknowns of the linear system obtained at each Picard iteration by a factor of the number of the space-time slab $E^{(t)}$, and enables the local resolution to be flexible; i.e., polynomial orders and grid resolutions can vary from one space-time slab to another.

To proceed, we approximate the integrals in (29) and (30) using Gaussian quadrature. Here, we choose the quadrature nodes to be the same as the interpolation nodes, i.e., Legendre–Gauss nodes, in both the spatial direction and the time direction. Such choice leads to an identity mass matrix in each space-time element.

In addition, the numerical flux \hat{u}_h^ϵ only depends on the variable u_h . So, the auxiliary variable $\boldsymbol{\sigma}_h$ can be eliminated from the formulation (30) by solving for $\boldsymbol{\sigma}_h$ in Eq. (29) and then substituting the resulting $\boldsymbol{\sigma}_h$ into Eq. (30). At the k th Picard iteration, we obtain a “space-time slab”, which is local to the slab in contrast to global to the whole time domain. The linear system for the space-time slab \mathcal{E}^n is,

$$\mathbf{H}(u_h^{(k)}) u_h^{(k+1)} = \mathbf{f}(u_h^{n-1}, u_{b,h}^k), \quad \mathbf{x} \in \mathcal{E}^n, \quad (31)$$

where $\mathbf{H}(u_h^{(k)})$ denotes the matrix of the linear system due to Eqs. (11) and (13), and the vector $\mathbf{f}(u_h^{n-1}, u_{b,h}^k)$ takes into account the solution at $\Omega^{t^{n-1}}$ computed from the previous space-time slab \mathcal{E}^{n-1} as well as the boundary conditions (see Fig. 2). Note that there is no boundary condition imposed on Ω^{t^n} .

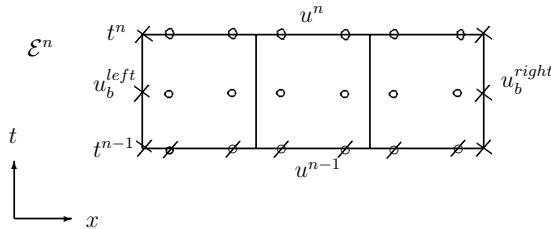


Fig. 2. Illustration of nodes distribution of the space-time slab $\mathcal{E}^n \in \mathbb{R}^2$, with the coordinates of a point $\bar{\mathbf{x}}$ denoted by $\bar{\mathbf{x}} = (x, t)$. In the spatial direction x , the total number of Legendre–Gauss points is 2, while 3 Legendre Gauss–Lobatto points are used in the time direction t . u_b^{left} and u_b^{right} denote the boundary data. The symbol “o” denotes the known values, u^{n-1} , while the symbol “x” denotes the boundary values, $u_{b,h}^k$. Values of u^{n-1} and $u_{b,h}^k$ form the vector $\mathbf{f}(u_h^{n-1}, u_{b,h}^k)$ in Eq. (31).

The order of matrix $\mathbf{H}(u_h^{(k)})$ depends on the polynomial order in space $p^{(x)}$, the polynomial order in time $p^{(t)}$, and the number of element $E^{(x)}$. For example, we solve the Burgers' equation in one spatial dimension, such that $f(u) = u^2/2$, and choose $p^{(x)} = 2$, $p^{(t)} = 2$ and $E^{(x)} = 2$. The order of the matrix $\mathbf{H}(u_h^{(k)})$ is (18×18) , i.e., $(p^{(x)} + 1) \times (p^{(t)} + 1) \times E^{(x)} = 18$, and the structure of the matrix $\mathbf{H}(u_h^{(k)})$ is a sparse nonsymmetric matrix.

Remark. Instead of a Picard iteration method, a Newton type method can be applied to solve the nonlinear system,

$$H(u)u = b. \quad (32)$$

One obtains the following by applying the Newton method,

$$\left(\frac{\partial f}{\partial u}\right)^k (u^{k+1} - u^k) = -H(u^k)u^k + b, \quad (33)$$

where $\frac{\partial f}{\partial u}$ is defined as follows:

$$\frac{\partial f}{\partial u} = u \frac{\partial H(u)}{\partial u} + H(u). \quad (34)$$

This method will be the same as the Picard iteration when $\frac{\partial H(u)}{\partial u}$ is zero. To avoid the evaluation of this derivative, we chose the Picard iteration rather than a Newton-type method.

5. Outline of the Space-Time DG Spectral Element Method

We present an outline of our space-time DG spectral element method combined with the shock tracking method for solving nonlinear hyperbolic problems in Algorithm 1. We illustrate our method based on the space-time formulation (21) in one spatial dimension. Note that $\tilde{\mathbf{B}}^{*,(k)}$ and $\tilde{\mathbf{A}}^{(k)}$ are nonlinear terms, such that,

$$\tilde{\mathbf{B}}^{*,(k)} = \tilde{\mathbf{B}}^{*,(k)}(\tilde{u}^{(k)}, W^{(k)}), \quad \tilde{\mathbf{A}}^{(k)} = \tilde{\mathbf{A}}(\tilde{u}^{(k)}). \quad (35)$$

6. Numerical Experiments

In this section, we test the space-time DG spectral element method on a variety of one dimensional problems to demonstrate the capabilities of our approach in preserving the accuracy and either capturing or tracking sharp shock profiles. Note that $\mathbf{H}(u_h^{(k)})$ in Eq. (31) is a nonsymmetric matrix due to the discretization of the convection term. We apply a direct solver from the LAPACK library in order to invert the resulting linear system. The tolerance of the Picard iteration is set to be 10^{-13} for the following tests.

6.1. Convergence analysis

In this test, the numerical results are compared with the exact solutions in order to examine the convergence rate with respect to both space and time. In the last

Algorithm 1 The space-time DG spectral element method combined with the shock tracking method for solving nonlinear hyperbolic problems in the space-time slab \mathcal{E}^{n+1} .

- 1: Given the solution u^n and the level set function ϕ^n at time t^n in the space-time slab \mathcal{E}^{n+1} with slab thickness Δt .
 - 2: Compute W^n at the discontinuity by the Rankine–Hugoniot jump condition.
 - 3: Define a space-time slab $\tilde{\mathcal{E}}^n$ in the moving frame of reference by introducing the backward transformation and the forward transformation.
 - 4: Compute the vector $\tilde{\mathbf{B}}^{*,n}$ based on \tilde{u}^n , and define $V^n = \partial f(\tilde{u}^n)/\partial \tilde{u}^n$;
 - 5: Compute the indicator S_K (14) and the matrix $\tilde{\mathbf{A}}^n$.
 - 6: Set $V^{j,(k)} = V^n$ and $W^{j,(k)} = W^n$, where $k = 0$ and $j = 1, \dots, p^{(t)}$. We initialize $\tilde{\mathbf{A}}$ as $\tilde{\mathbf{A}}^{0,(k)} = \tilde{\mathbf{A}}^n$ and $\tilde{\mathbf{A}}^{j,(k)} = 0$. Note that $V^{p^{(t)},(k)} = V^{n+1,(k)}$, $W^{p^{(t)},(k)} = W^{n+1,(k)}$ and $\tilde{\mathbf{A}}^{p^{(t)},(k)} = \tilde{\mathbf{A}}^{n+1,(k)}$.
 - 7: **for** $k_s = 1, 2$ **do**
 - (a) **while** ($\|V^{n+1,(k+1)} - V^{n+1,(k)}\| > tol$) **do**
 - (i) Use the DG spectral element method to discretize (21).
 - (ii) Solve the resulting space-time slab linear system (31).
 - (iii) Compute $V^{j,(k+1)}$ ($j = 1, \dots, p^{(t)}$).
 - (iv) Update $W^{j,(k+1)}$ at $j = 1, \dots, p^{(t)}$, and $\tilde{\mathbf{B}}^{*,(k+1)}$.
 - (b) **end while**
 - (c) Compute V^j and W^j at $j = 1, \dots, p^{(t)}$, and then update $\tilde{\mathbf{B}}^*$.
 - (d) Compute the indicator S_K (14) and $\tilde{\mathbf{A}}^{j,(k+1)}$ at $j = 1, \dots, p^{(t)}$.
 - 8: **end for**
 - 9: Compute $\phi^{p^{(t)},(k)}$ by the forward transformation, and $u^{p^{(t)},(k+1)}$ by the backward transformation.
 - 10: **return** u^{n+1} and ϕ^{n+1} .
-

space-time slab $\mathcal{E}^{E(t)}$, the errors are measured in the L_∞ norm at time $t^{E(t)} = T$, where T is the final computational time.

$$\|Err_u\|_\infty = \max_{i=0, \dots, p^{(x)}} \|u_i^{E(t)} - (u_h)_i^{E(t)}\|, \quad (36)$$

where $u_i^{E(t)}$ denotes the exact solution evaluated at node x_i , and $(u_h)_i^{E(t)}$ is the approximation evaluated at the same node.

We consider the Burgers' equation

$$u_t + \left(\frac{u^2}{2}\right)_x = (\epsilon u_x)_x, \quad x \in [-1.0, 1.0], \quad (37)$$

with an exact traveling wave solution

$$u(x, t) = -\tanh\left(\frac{x + 0.5 - t}{2\epsilon}\right) + 1.0. \quad (38)$$

Dirichlet boundary conditions are imposed by using the exact solution. We refer the reader to [Furbish *et al.* (2008)] and the references therein for deriving an exact solution of the Burgers' equation. We set $\epsilon = 0.5$ in the following tests.

First, we demonstrate the spectral accuracy of the approximation by plotting the maximum errors in the solution as a function of the spatial polynomial order $p^{(x)}$. The simulation is computed over the time $t = 0$ to $t = 1.0$, and the comparison is made among different numbers of spatial elements, i.e., $E^{(x)} = 5$ and $E^{(x)} = 10$. The ratio of the time step to the spatial element size is $\Delta t/\Delta x = 0.125$ for $E^{(x)} = 5$ and $\Delta t/\Delta x = 0.25$ for $E^{(x)} = 10$. On the left part of Fig. 3, the polynomial order in the temporal direction $p^{(t)}$ is chosen to be the same as $p^{(x)}$, while $p^{(t)}$ is chosen to be 12 so that the temporal errors are negligible on the right part. Both choices of $p^{(t)}$ exhibit spectral accuracy in space.

Next, we demonstrate the spectral accuracy of the approximation by plotting the maximum errors in the solution as a function of the temporal polynomial order $p^{(t)}$. The simulation is computed over the time $t = 0$ to $t = 1.0$, and the spatial domain is divided into five elements, i.e., $E^{(x)} = 5$. The comparison is made among $E^{(t)} = 10$ and $E^{(t)} = 20$. The ratio of the time step to the spatial element size is,

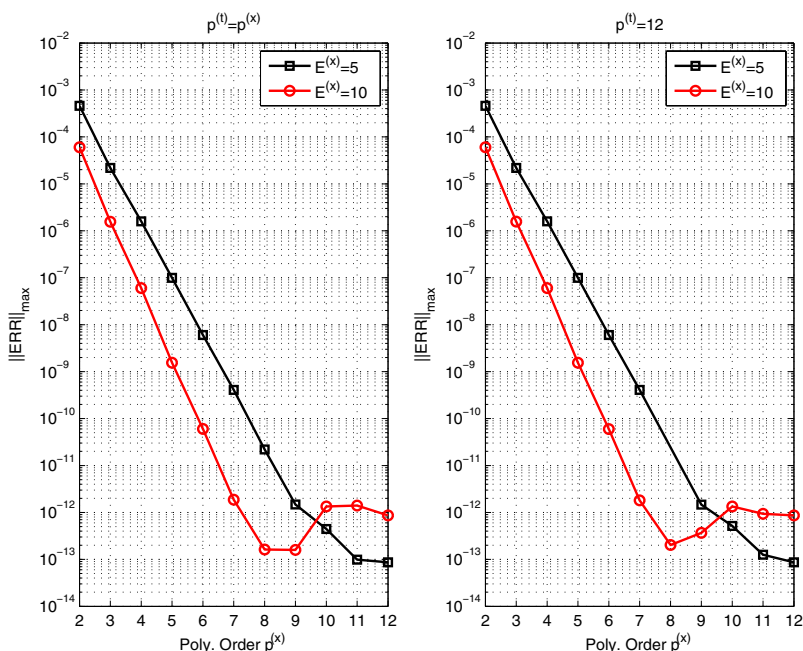


Fig. 3. Errors in the solution as a function of polynomial order $p^{(x)}$. The polynomial order in the temporal direction $p^{(t)}$ is chosen to be the same as $p^{(x)}$ on the left part, and to be 12 on the right part. The comparison is made between $E^{(x)} = 5$ and $E^{(x)} = 10$. The ratio of the time step to the spatial element size is, $\Delta t/\Delta x = 0.125$ for $E^{(x)} = 5$ and $\Delta t/\Delta x = 0.25$ for $E^{(x)} = 10$. The simulation is computed over the time $t = 0$ to $t = 1.0$.

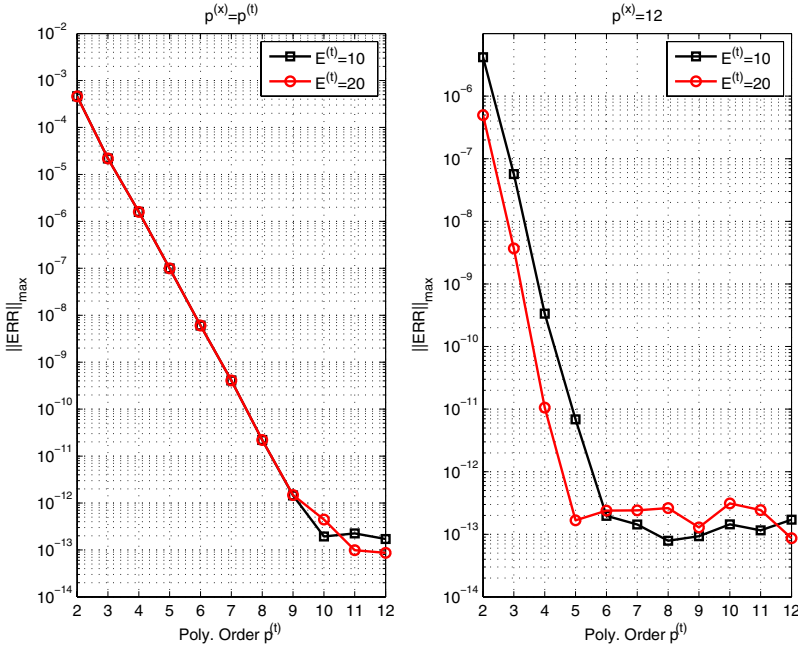


Fig. 4. Errors in the solution as a function of polynomial order $p^{(t)}$. The polynomial order in the spatial direction $p^{(x)}$ is chosen to be the same as $p^{(t)}$ on the left part, and to be 12 on the right part. The comparison is made among $E^{(t)} = 10$ and $E^{(t)} = 20$. The ratio of the time step to the spatial element size is, $\Delta t/\Delta x = 0.25$ for $E^{(t)} = 10$ and $\Delta t/\Delta x = 0.125$ for $E^{(t)} = 20$. The simulation is computed over the time $t = 0$ to $t = 1.0$, and $E^{(x)} = 5$.

$\Delta t/\Delta x = 0.25$ for $E^{(t)} = 10$ and $\Delta t/\Delta x = 0.125$ for $E^{(t)} = 20$. On the left part of Fig. 4, the polynomial order in the spatial direction $p^{(x)}$ is chosen to be the same as $p^{(t)}$, while $p^{(x)}$ is chosen to be 12 so that the spatial errors are negligible on the right part. Both choices of $p^{(x)}$ exhibit spectral accuracy in time.

Remark. Since the proposed space-time spectral element method is fully implicit, there is no Courant–Friedrichs–Lewy (CFL) restriction. If the space-time spectral element method was implemented with an explicit time integration scheme [Pei *et al.* (2017)], then the CFL condition is as follows:

$$\max_{m,i} |u|_{m,i} \frac{\max_m \{\Delta t_m\}}{\min_i \{\Delta x_i\}} = \max_{m,i} |u|_{m,i} \frac{\max_m \{t_{m+1}^n - t_m^n\}}{\min_i \{\Delta x/2(\xi_{i+1} - \xi_i)\}} < 1, \quad (39)$$

where $\{t_m^n\}_{m=0,\dots,p^{(t)}}$ is the set of Legendre Gauss–Lobatto points illustrated in Fig. 1, and $\{\xi_i\}_{i=0,\dots,p^{(x)}}$ is the set of Legendre Gauss–Lobatto points defined on $[-1, 1]$. As the polynomial order $p^{(x)}$ is increased, the denominator will be proportional to $1/(p^{(x)})^2$. Thus, the explicit CFL condition (39) is more stringent than the values used for $(\Delta t/\Delta x)$ when the spatial order is large and the temporal order is small. More discussion can be found in Table 1.

Table 1. Dependence of the number of Picard iterations on the time step and the temporal order for the test problem of Eqs. (40) and (41). The criterion for convergence of the Picard iteration is $\text{tol} = 10^{-13}$. The maximum iteration number N_{\max} is set to be 1000. If the last column $\max_m \{\Delta t_m\} / \min_i \{\Delta x_i\}$ is greater than $\frac{1}{\max |u|} = 2/3$, then the CFL condition would have been violated if an explicit method was used. See Eq. (39).

$E^{(x)}$	$p^{(x)}$	$p^{(t)}$	$\Delta t / \Delta x$	Number of iterations	$\max_m \{\Delta t_m\} / \min_i \{\Delta x_i\}$
8	8	1	0.08	65	1.59
8	8	2	0.08	47	0.79
8	8	3	0.08	33	0.71
8	8	4	0.08	25	0.52
8	8	5	0.08	24	0.45
8	8	6	0.08	22	0.37
8	8	7	0.08	19	0.33
8	8	1	0.2	$> N_{\max} (*)$	3.99
8	8	2	0.2	$> N_{\max} (*)$	1.99
8	8	3	0.2	124	1.78
8	8	4	0.2	51	1.30
8	8	5	0.2	46	1.13
8	8	6	0.2	40	0.93
8	8	7	0.2	33	0.83

6.2. The spatio-temporal artificial viscosity

In order to feature the gains of our artificial diffusion approach, we test our approach on a linear advection problem with periodic boundary conditions as follows:

$$\begin{cases} u_t + u_x = 0, & -1 \leq x \leq 1, \quad t > 0, \\ u(x, 0) = u_0(x). \end{cases}$$

The initial condition is chosen to be

$$u_0(x) = \begin{cases} 1.0, & \text{if } x \in [-1.0, -0.5), \\ \sin(\pi(x + 0.5)), & \text{if } x \in [-0.5, 0.5), \\ 1.0, & \text{if } x \in [0.5, 1.0], \end{cases}$$

We compute the solutions without adding any artificial viscosity and show the comparisons between the numerical solutions and the exact solutions at time $t = 2.0$ in Fig. 5. Different pairs of polynomial orders are also compared: $(p^{(x)}, p^{(t)}) = (10, 5)$ (on the left) and $(p^{(x)}, p^{(t)}) = (10, 7)$ (on the right). The results obtained by a polynomial with a higher temporal order, $p^{(t)} = 7$, are the same as the one computed by a lower temporal order, $p^{(t)} = 5$.

Now, we demonstrate the performance of adding artificial viscosity. Initially, in a given space-time slab, the known information using to compute the amount of artificial viscosity is only available on the bottom of the space-time slab. In order to obtain the amount of artificial viscosity in the whole space-time slab, a *prediction-correction* scheme is implemented. The simulation is computed over the time $t = 0$ to $t = 2.0$, and $E^{(x)} = 10$. The ratio of the time step to the element size $\Delta t / \Delta x = 0.25$. For the empirical tuning parameters in (15), we choose $\kappa = 2.5$ and

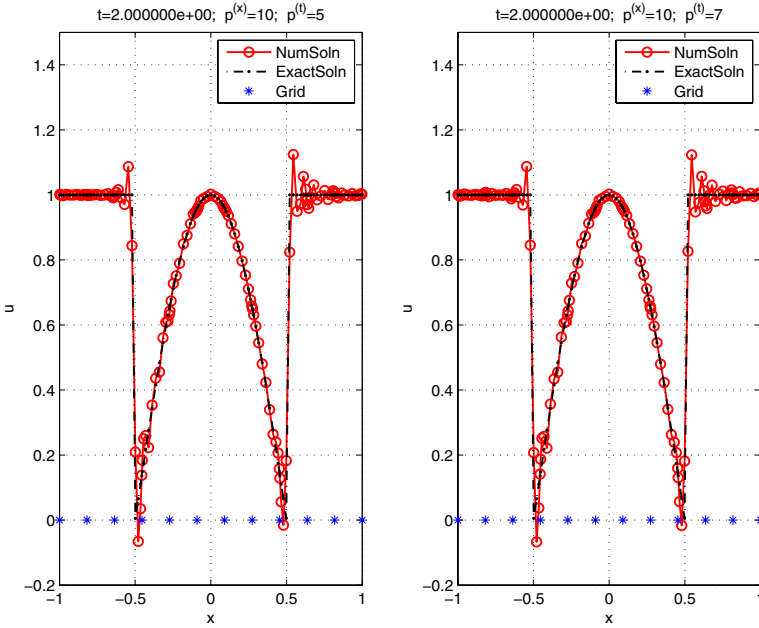


Fig. 5. A comparison between the numerical results, without any artificial viscosity, and exact solutions using different pairs of polynomial orders: $(p^{(x)}, p^{(t)}) = (10, 5)$ (on the left) and $(p^{(x)}, p^{(t)}) = (10, 7)$ (on the right). The simulation is computed over the time $t = 0$ to $t = 2.0$, and $E^{(x)} = 11$. The ratio of the time step to the element size $\Delta t/\Delta x = 0.055$.

$\epsilon_0 = 0.25\Delta x/p^{(x)}$. In Fig. 6, the comparison between the numerical solution and the exact solution are plotted on the left, while the amount of artificial viscosity for the last time level in the space-time slab is plotted on the right. Clearly, the oscillations are reduced by adding artificial viscosity.

Note that the amount of artificial viscosity plotted in Fig. 6 is not symmetric. The plot of the magnitude of artificial viscosity in Fig. 6 is not symmetric because this result is plotted at the end time after the wave has already completed one full revolution about the periodic domain. Asymmetries are developed in the intermediate time steps when the discontinuities do not align symmetrically in respective elements and then these asymmetries are propagated along characteristics.

In Fig. 7, we show the time-varying artificial viscosity in the last space-time slab. The amount of artificial viscosity is varying amongst the different time “nodes” in a given space-time slab as the discontinuities travel along characteristics.

6.3. Shock capturing

In this section, we study the sensitivity of the error and the number of Picard iterations on varying the temporal and spatial orders, the number of elements, the ratio of the time step to the element size, and whether or not artificial viscosity is

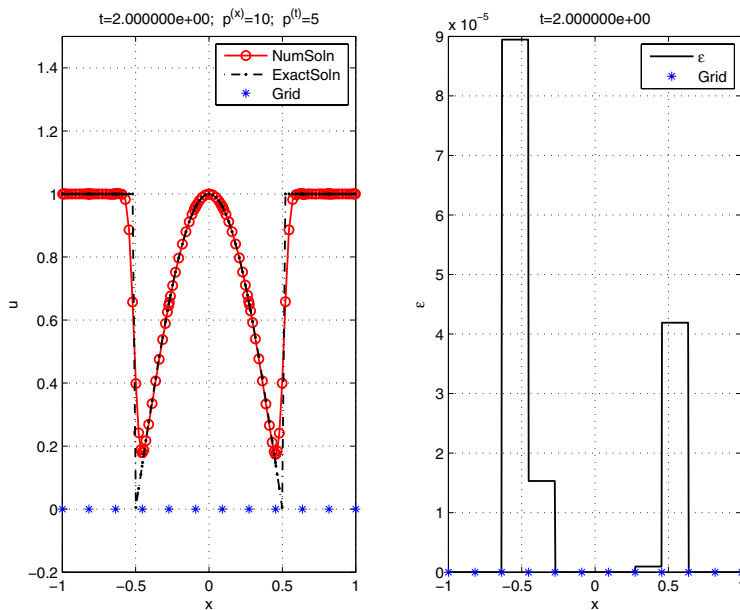


Fig. 6. A comparison between the numerical results and exact solutions using $(p^{(x)}, p^{(t)}) = (10, 5)$. On the left, comparisons between the numerical solution and the exact solution are plotted, while the amount of artificial viscosity is plotted on the right.

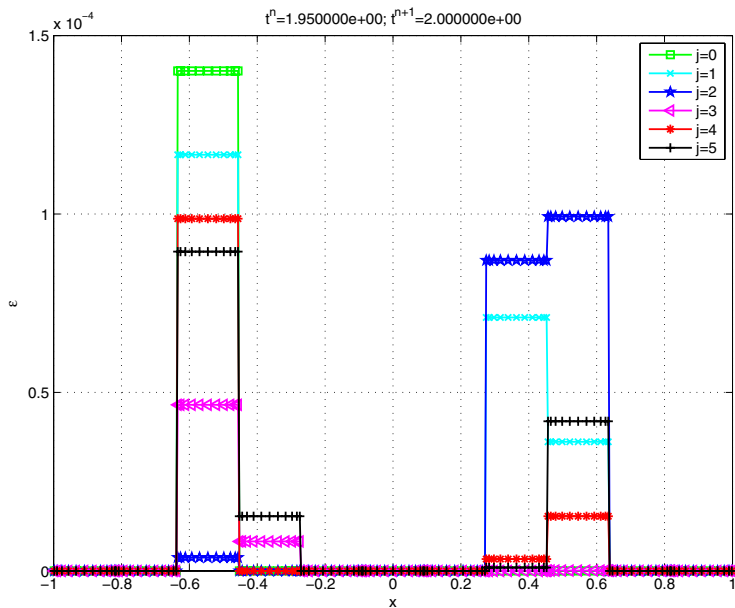


Fig. 7. A plot of the time-varying artificial viscosity for an approximation with $(p^{(x)}, p^{(t)}) = (10, 5)$ in the last space-time slab. The amount of artificial viscosity is plotted for each time level (node) j in the last time slab.

present. For the sensitivity study, the inviscid Burgers' equation is solved,

$$u_t + \partial f(u)/\partial x = 0, \quad 0 \leq x \leq 1, \quad t > 0, \quad (40)$$

$$u(x, 0) = \frac{1}{2} + \sin(2\pi x). \quad (41)$$

Periodic boundary conditions are enforced at $x = 0$ and $x = 1$. The solution of this problem forms a shock at $t = 0.25$ that moves to the right. The empirical tuning parameters in Eq. (15) is set to be $\kappa = 2.5$ and $\epsilon_0 = \Delta x/p^{(x)}$.

First, we report the sensitivity of the number of Picard iterations as a function of the temporal order and the ratio of the time step to the spatial element size in Table 1. The simulation is computed over the time $t = 0.0$ to $t = 0.25$. The number of Picard iterations, reported in Table 1, represents the maximum number during the simulation process. As illustrated in the table, approximations with higher temporal orders reduce the number of Picard iterations for this test. Note that the Picard iteration algorithm does not converge within the maximum iteration number (one thousand) in the prediction step for $p^{(t)} = 1$ and $p^{(t)} = 2$ (noted by $(*)$ in Table 1). However, the Picard iteration in the correction step converges in 163 ($p^{(t)} = 1$) or 164 ($p^{(t)} = 2$) iterations. We reason for this is because the prediction step, in spite of not converging, yielded an artificial viscosity that was sufficient to damp out noise

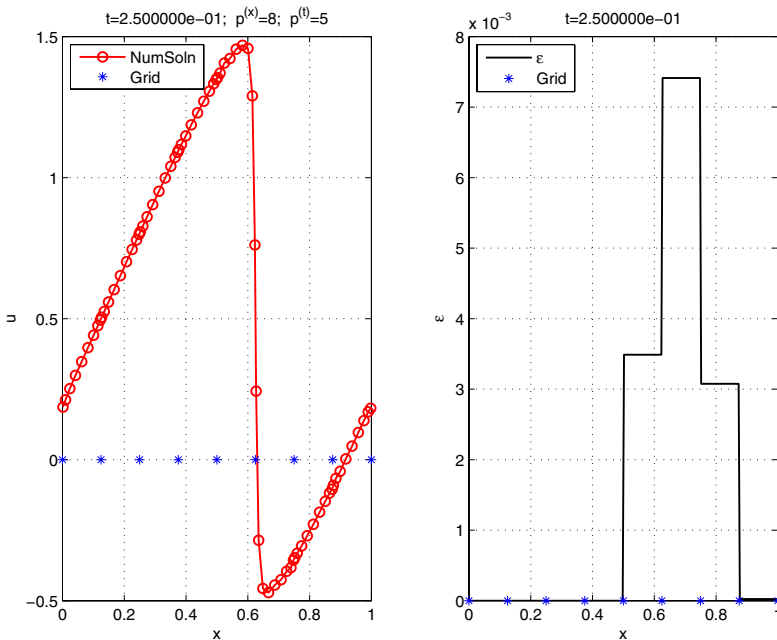


Fig. 8. A plot of the approximate solution at $t = 0.25$ (on the left) and the amount of artificial viscosity (on the right). The spatial polynomial order is $p^{(x)} = 8$ and the temporal polynomial order is $p^{(t)} = 5$. The number of spatial elements is $E^{(x)} = 8$. The ratio of the time step to the spatial element size is set as $\Delta t/\Delta x = 0.08$.

for carrying out the correction step. Note that the number of Picard iterations is determined by the criterion for convergence of the Picard method, and we choose the criterion to be $\text{tol} = 10^{-13}$.

In order to demonstrate the fully implicit property, we also list the ratio of temporal step to spatial step, $\max_m\{\Delta t_m\}/\min_i\{\Delta x_i\}$, that should be used for an explicit time integration in Table 1. The term of $\max_m\{\Delta t_m\}/\min_i\{\Delta x_i\}$ can be derived from Eq. (39), and the valid values should be less than $1/\max|u| = 0.67$ in this test. Clearly, most of them are failed in our test. If an explicit time integration is used, then one should choose smaller time step than the one used in this paper.

In Figs. 8 and 9, the numerical solutions (on the left) and the amount of artificial viscosity (on the right) are plotted. The simulation is computed over the time $t = 0.0$ to $t = 0.25$. The ratio of the time step to the spatial element size is set as $\Delta t/\Delta x = 0.08$. In addition, the shock is well-resolved for approximations with different higher orders approximations, i.e., $(p^{(x)}, p^{(t)}) = (8, 5)$ and $(p^{(x)}, p^{(t)}) = (13, 7)$. In comparing Figs. 8–9, an approximation with higher order, $(p^{(x)}, p^{(t)}) = (13, 7)$, resolves the solution extrema more accurately than the one with $(p^{(x)}, p^{(t)}) = (8, 5)$ (Note that the same number of spatial elements are used).

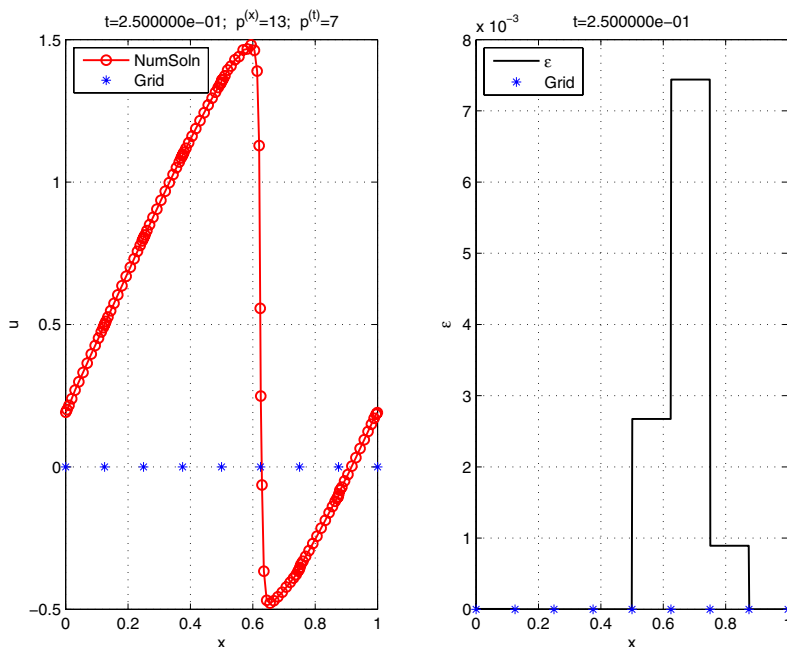


Fig. 9. A plot of the approximate solution at $t = 0.25$ (on the left) and the amount of artificial viscosity (on the right). The spatial polynomial order is $p^{(x)} = 13$ and the temporal polynomial order is $p^{(t)} = 7$. The number of spatial elements is $E^{(x)} = 8$. The ratio of the time step to the spatial element size is set as $\Delta t/\Delta x = 0.08$.

6.4. Shock tracking

The inviscid Burgers' equation with periodic boundary conditions is solved with the following initial conditions,

$$u(x, 0) = \begin{cases} \sin(2\pi(x + 0.25)) + 0.25, & -0.25 \leq x \leq 0, \\ -\cos(2\pi x) + 0.25, & 0 < x \leq 0.25. \end{cases}$$

Since the location of the discontinuity is known at the initial time, we track the discontinuity explicitly. In this section, we report the error of our tracking algorithm with respect to the spatial and temporal polynomial orders. We demonstrate the spectral accuracy of the shock velocity and shock location by plotting the maximum errors as a function of polynomial orders $p^{(x)}$ and $p^{(t)}$. The errors are computed by comparing our numerical results with the ones computed by a first-order Godunov method with 10,000 grid points. Since the temporal resolution affects the number of Picard iterations, we choose the time step for the convergence tests to be 0.01 so that the ratio of the time step to the spatial element size ($\Delta t/\Delta x$) is 0.04 for $E^{(x)} = 2$ and 0.06 for $E^{(x)} = 3$. This setting for Δt allows the Picard iteration algorithm to converge for a lower temporal order, e.g., $p^{(t)} = 1$.

In Fig. 10, we choose the number of spatial elements to be $E^{(x)} = 3$ (on the left) and $E^{(x)} = 2$ (on the right). For the case of $E^{(x)} = 2$, the discontinuity is

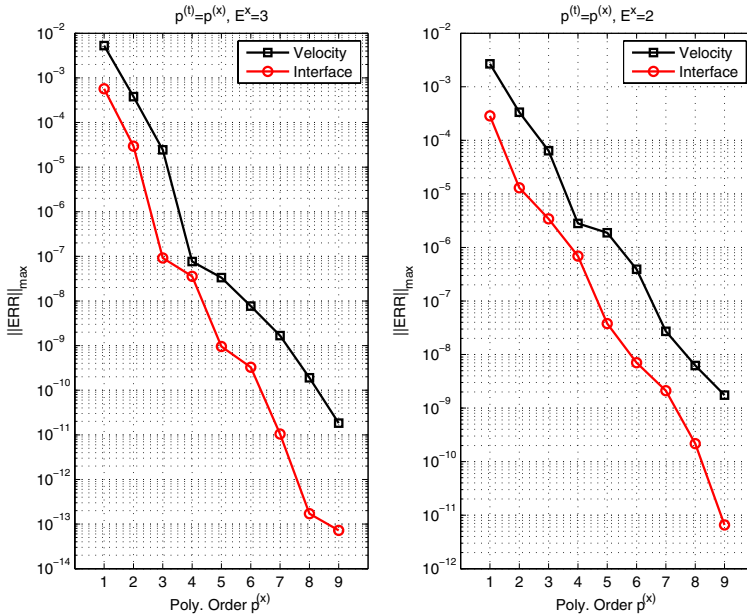


Fig. 10. Errors in the shock velocity and shock location as a function of polynomial order $p^{(x)}$. The polynomial order in the temporal direction $p^{(t)}$ is chosen to be the same as $p^{(x)}$. The number of elements are chosen to be $E^{(x)} = 3$ (on the left) and $E^{(x)} = 2$ (on the right). The simulation is computed over the time $t = 0$ to $t = 0.2$. The ratio of the time step to the spatial element size is $\Delta t/\Delta x = 0.06$ for $E^{(x)} = 3$ and $\Delta t/\Delta x = 0.04$ for $E^{(x)} = 2$.

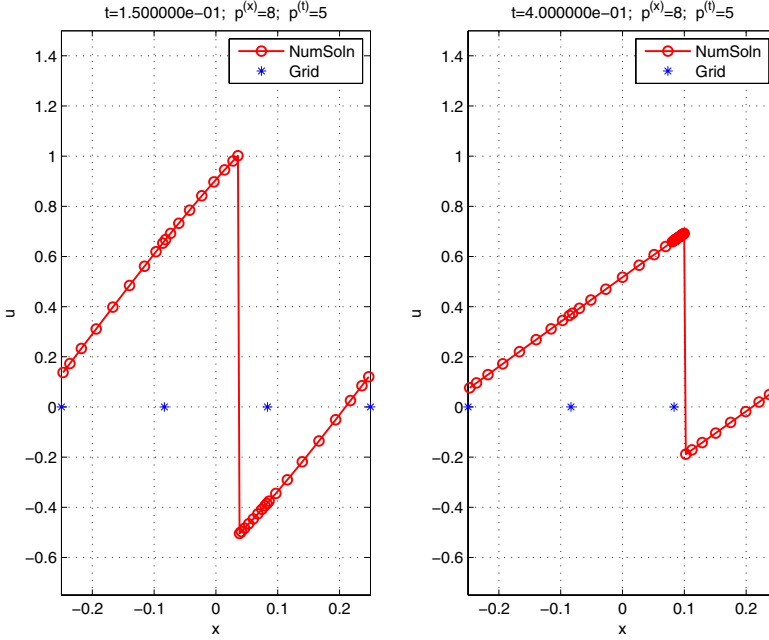


Fig. 11. A plot of the numerical solution at different times, especially involving dealing with the small cut-cell. The ratio of the time step to the spatial element size is $\Delta t/\Delta x = 0.06$, and $E^{(x)} = 3$. The polynomial orders are set to be $(p^{(x)}, p^{(t)}) = (8, 5)$.

initialized at the inter-element boundary, which means that there are no cut-cells at the beginning of the simulation. After the first time step, a small cell is generated on the left side of the discontinuity so that $\Delta t/\min_i\{\Delta x_i\}$ could be greater than 1. For such scenarios, we refine the time step in the given space-time slab. We show that the small cell does not affect the spectral accuracy in our tracking method. Both the interface velocity and the interface location exhibit spectral accuracy in both space and time.

In Fig. 11, the approximation with $(p^{(x)}, p^{(t)}) = (8, 5)$ behaves well after crossing an inter-element boundary.

7. Summary and Conclusions

A space-time DG spectral element method for solving nonlinear hyperbolic problems has been presented in one spatial dimension. In order to treat discontinuous solutions of hyperbolic problems, two methods are introduced in the space-time framework: (i) the shock capturing approach and (ii) the shock tracking approach.

Numerical experiments on the Burgers' equation demonstrate the spectral convergence in both space and time. The spatio-temporal artificial diffusion approach is tested on both a scalar hyperbolic linear problem with a discontinuous initial condition and the inviscid Burgers' equation with a solution that evolves into a

sharp discontinuity. In the present shock capturing method, the amount of artificial viscosity is computed at each time level (node) in a given space-time slab based on the smoothness indicator, which results in optimal diffusion to suppress Gibbs phenomenon. Space-time spectral accuracy enables better accuracy at extrema away from the shock, and enables better accuracy at capturing the shock strength at a shock with higher order polynomials in both space and time compared to lower order polynomials. Simultaneously, Gibbs oscillations are suppressed by the combination of higher order polynomials and the present shock capturing scheme. The sensitivity of the number of Picard iterations as a function of the temporal order with different ratios of the time step to the spatial element size is discussed. It has been found that increasing the temporal order reduces the number of Picard iterations in the nonlinear case and eliminates oscillations in the linear case.

The spectral accuracy of the shock speed and location is demonstrated for the shock tracking method. Also, the space-time spectral accuracy of the solution on either side of the discontinuity is preserved. A small cut-cell may be generated in the shock tracking process, which does not affect the spectral accuracy. In addition, higher order approximations, e.g., $(p^x, p^{(t)}) = (6, 6)$ and $(p^x, p^{(t)}) = (8, 5)$, behave well for the case in which a small cell is generated in the tracking process. However, a drawback of the tracking method is the time step restriction caused by a tiny cut-cell. A tiny cut-cell can result in the collision of characteristics for the nonlinear problem; this problem is manifested in the Picard iteration not converging unless the time step is sufficiently refined. The ratio of the time step to the spatial element size $\Delta t / \min_i \{\Delta x_i\}$ must be restricted depending on $\max\{\partial f(u)/\partial u\}$ if a tiny cut-cell is generated in one space-time slab. An alternate to reduce Δt is to adjust the polynomial order used in a given element. The capability for subelements in a cut-cell to have a low order approximation that is distinct from its neighbors is the topic of future research.

The present method can be extended to higher spatial dimensions, however, an alternative faster direct solver methodology, such as Martinsson [2013], needs to be explored for the resulting large sparse linear system (31) obtained in each Picard iteration. This will also be the subject of future study.

Acknowledgments

This work was partially supported by the National Science Foundation under contract DMS 1418983.

References

- Abbassi, H., Mashayek, F. and Jacobs, G. B. [2014] “Shock capturing with entropy-based artificial viscosity for staggered grid discontinuous spectral element method,” *Computers and Fluids* **98**, 152–163, [12th USNCCM mini-symposium of High-Order Methods for Computational Fluid Dynamics — A special issue dedicated to the 80th birthday of Professor Antony Jameson.]

- Arnold, D. N., Brezzi, F., Cockburn, B. and Marini, L. D. [2001] "Unified analysis of discontinuous Galerkin methods for elliptic problems," *SIAM J. Numer. Anal.* **39**(5), 1749–1779.
- Balsara, D. S. and Shu, C.-W. [2000] "Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy," *J. Comput. Phys.* **160**(2), 405–452.
- Benzi, M. and Olshanskii, M. A. [2006] "An augmented Lagrangian-based approach to the Oseen problem," *SIAM J. Sci. Comput.* **28**(6), 2095–2113.
- Canuto, C., Hussaini, M. Y., Quarteroni, A. and Zang, T. A. [2006] *Spectral methods: Fundamentals in Single Domains* (Springer, Berlin, 2006).
- Cockburn, B. B., Karniadakis, G. and Shu, C.-W. (eds.) [2000] *Discontinuous Galerkin methods: theory, computation, and applications*, Lecture notes in computational science and engineering (Springer, Berlin, New York, 2000).
- Cockburn, B., Kanschat, G., Perugia, I. and Schötzau, D. [2001] "Superconvergence of the local discontinuous Galerkin method for elliptic problems on Cartesian grids," *SIAM J. Numer. Anal.* **39**(1), 264–285.
- Cockburn, B. and Shu, C.-W. [1998] "The local discontinuous Galerkin method for time-dependent convection-diffusion systems," *SIAM J. Numer. Anal.* **35**(6), 2440–2463.
- Daubechies, I. [1992] *Ten Lectures on Wavelets*, CBMS-NSF regional Conference Series in Applied Mathematics, Vol. 61 (SIAM, Philadelphia, Pennsylvania).
- Duraisamy, K., Baeder, J. D. and Liu, J.-G. [2003] "Concepts and application of time-limiters to high resolution schemes," *J. Sci. Comput.* **19**(1–3), 139–162.
- Furbish, D., Yussuf Hussaini, M., Dimet, F.-X. L., Ngnepieba, P. and Wu, Y. [2008] "On discretization error and its control in variational data assimilation" *Tellus A* **60**(5), 979–991.
- Gerritsen, M. and Olsson, P. [1996] "Designing an efficient solution strategy for fluid flows: 1. a stable high order finite difference scheme and sharp shock resolution for the euler equations," *J. Comput. Phys.* **129**(2), 245–262.
- Gustafsson, B. and Olsson, P. [1995] "Fourth-order difference methods for hyperbolic ibvps," *J. Comput. Phys.* **117**(2), 300–317.
- Harten, A. [1978] "The artificial compression method for computation of shocks and contact discontinuities. iii. self-adjusting hybrid schemes," *Math. Comput.* **32**(142), 363–389.
- Hesthaven, J. S. and Warburton, T. [2008] *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, Vol. 54, *Texts in Applied Mathematics* (Springer, New York).
- Jiang, G.-S. and Shu, C.-W. [1996] "Efficient implementation of weighted eno schemes," *J. Comput. Phys.* **126**(1), 202–228.
- Kadioglu, S. [2011] "A gas dynamics method based on the spectral deferred corrections (sdc) time integration technique and the piecewise parabolic method (ppm)" *Am. J. Comput. Math.* **1**(4), 303–317.
- Kadioglu, S. Y. [2016] "An essentially nonoscillatory spectral deferred correction method for hyperbolic problems," *Int. J. Comput. Method* **13**(03), 1650017.
- Karniadakis, G. E. and Sherwin, S. J. [2013] *Spectral/hp element methods for CFD*, Numerical Mathematics and Scientific Computation, 2nd Edition (Oxford University Press, Oxford).
- Klaij, C. M., van der Vegt, J. J. W. and van der Ven, H. [2006] "Space-time discontinuous Galerkin method for the compressible Navier–Stokes equations," *J. Comput. Phys.* **217**(2), 589–611.

- Kopriva, D. A. [2009] *Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers*. Scientific Computation (Springer, Berlin).
- Layton, A. T. and Minion, M. L. [2004] "Conservative multi-implicit spectral deferred correction methods for reacting gas dynamics," *J. Comput. Phys.* **194**(2), 697–715.
- Lörcher, F., Gassner, G. and Munz, C.-D. [2007] "A discontinuous Galerkin scheme based on a space-time expansion. I. Inviscid compressible flow in one space dimension," *J. Sci. Comput.* **32**(2), 175–199.
- Mallat, S. [1999] *A Wavelet Tour of Signal Processing* (Academic Press, London).
- Mallat, S. and Zhong, S. [1992] "Characterization of signals from multiscale edges," *IEEE Transactions on Pattern Analysis & Machine Intelligence* **14**(7), 710–732.
- Martinsson, P. G. [2013] "A direct solver for variable coefficient elliptic PDEs discretized via a composite spectral collocation method," *J. Comput. Phys.* **242**, 460–479.
- Mavriplis, C. [1994] "Adaptive mesh strategies for the spectral element method," *Comput. Methods Appl. Mech. Engrg.* **116**(1–4), 77–86.
- Pei, C., Sussman, M. and Hussaini, M. Y. [2017] "A space-time discontinuous galerkin spectral element method for the stefan problem," *Discrete and Continuous Dynamical Systems — Series B*, doi: 10.3934/dcdsb.2017216.
- Pei, C., Sussman, M. and Hussaini, Y. M. [2017] "New multi-implicit space-time spectral element methods for advection-diffusion-reaction problems," *J. Sci. Comput.*, doi: 10.1007/s10915-018-0654-5.
- Persson, P. and Peraire, J. [2006] "Sub-cell shock capturing for discontinuous Galerkin methods, *AIAA Aerospace Sciences Meeting and Exhibit*, 44th *AIAA Aerospace Sciences Meeting and Exhibit*, 2006 Jan 9, p. 112.
- Pirozzoli, S. [2002] "Conservative hybrid compact-weno schemes for shock-turbulence interaction," *J. Comput. Phys.* **178**(1), 81–117.
- Reisner, J., Serencsa, J. and Shkoller, S. [2013] "A space-time smooth artificial viscosity method for nonlinear conservation laws," *J. Comput. Phys.* **235**, 912–933.
- Ren, Y.-X., Liu, M. and Zhang, H. [2003] "A characteristic-wise hybrid compact-weno scheme for solving hyperbolic conservation laws," *J. Comput. Phys.* **192**(2), 365–386.
- Rhebergen, S. and Cockburn, B. [2012] "A space-time hybridizable discontinuous Galerkin method for incompressible flows on deforming domains," *J. Comput. Phys.* **231**(11), 4185–4204.
- Rhebergen, S., Cockburn, B. and van der Vegt, J. J. W. [2013] "A space-time discontinuous Galerkin method for the incompressible Navier–Stokes equations," *J. Comput. Phys.* **233**, 339–358.
- Shu, C.-W. and Osher, S. [1989] "Efficient implementation of essentially non-oscillatory shock-capturing schemes, ii," *J. Comput. Phys.* **83**(1), 32–78.
- Sjögreen, B. and Yee, H. C. [2004] "Multiresolution wavelet based adaptive numerical dissipation control for high order methods," *J. Sci. Comput.* **20**(2), 211–255.
- Sollie, W. E. H., Bokhove, O. and van der Vegt, J. J. W. [2011] "Space-time discontinuous Galerkin finite element method for two-fluid flows," *J. Comput. Phys.* **230**(3), 789–817.
- Sudirham, J. J., van der Vegt, J. J. W. and van Damme, R. M. J. [2006] "Space-time discontinuous Galerkin method for advection-diffusion problems on time-dependent domains," *Appl. Numer. Math.* **56**(12), 1491–1518.
- Suresh, A. and Huynh, H. T. [1997] "Accurate monotonicity-preserving schemes with runge–kutta time stepping," *J. Comput. Phys.* **136**(1), 83–99.
- Sussman, M. and Hussaini, M. Y. [2003] "A discontinuous spectral element method for the level set equation," *J. Sci. Comput.* **19**(1–3), 479–500. Special issue in honor of the sixtieth birthday of Stanley Osher.

- Sussman, M., Smereka, P. and Osher, S. [1994] “A level set approach for computing solutions to incompressible two-phase flow,” *J. Comput. Phys.* **114**(1), 146–159.
- Touil, H., Hussaini, M. Y. and Sussman, M. [2007] “Tracking discontinuities in hyperbolic conservation laws with spectral accuracy,” *J. Comput. Phys.* **225**(2), 1810–1826.
- van der Vegt, J. J. W. and Sudirham, J. J. [2008] “A space-time discontinuous Galerkin method for the time-dependent Oseen equations,” *Appl. Numer. Math.* **58**(12), 1892–1917.
- Yee, H. C., Sandham, N. D. and Djomehri, M. J. [1999] “Low-dissipative high-order shock-capturing methods using characteristic-based filters,” *J. Comput. Phys.* **150**(1), 199–238.