Commun. Comput. Phys.
doi: 10.4208/cicp.050309.110510a

Vol. 9, No. 2, pp. 441-480
February 2011

A Compact High Order Space-Time Method for Conservation Laws

Shuangzhang Tu*, Gordon W. Skelton and Qing Pang

Department of Computer Engineering, Jackson State University, Jackson, MS 39217, USA.

Received 5 March 2009; Accepted (in revised version) 11 May 2010

Communicated by Chi-Wang Shu

Available online 27 August 2010

Abstract. This paper presents a novel high-order space-time method for hyperbolic conservation laws. Two important concepts, the staggered space-time mesh of the space-time conservation element/solution element (CE/SE) method and the local discontinuous basis functions of the space-time discontinuous Galerkin (DG) finite element method, are the two key ingredients of the new scheme. The staggered space-time mesh is constructed using the cell-vertex structure of the underlying spatial mesh. The universal definitions of CEs and SEs are independent of the underlying spatial mesh and thus suitable for arbitrarily unstructured meshes. The solution within each physical time step is updated alternately at the cell level and the vertex level. For this solution updating strategy and the DG ingredient, the new scheme here is termed as the discontinuous Galerkin cell-vertex scheme (DG-CVS). The high order of accuracy is achieved by employing high-order Taylor polynomials as the basis functions inside each SE. The present DG-CVS exhibits many advantageous features such as Riemann-solver-free, high-order accuracy, point-implicitness, compactness, and ease of handling boundary conditions. Several numerical tests including the scalar advection equations and compressible Euler equations will demonstrate the performance of the new method.

AMS subject classifications: 65M99, 76M25

Key words: High order method, space-time method, cell-vertex scheme (CVS), conservation laws.

1 Introduction

To numerically solve hyperbolic conservation laws, many methods including the classic finite difference and finite volume methods, discontinuous Galerkin (DG) method [1, 6,

*Corresponding author. *Email addresses:* shuangzhang.tu@jsums.edu (S. Tu), gordon.skelton@jsums.edu (G. W. Skelton), qing.pang@jsums.edu (Q. Pang)

9, 17] and the spectral volume method [31] use exact or approximate Riemann solvers to provide inter-cell fluxes. Since Riemann fluxes are based on the local characteristic structure of the governing equations, they are expected to provide the correct fluxes and stabilize the numerical methods. However, in the actual implementation, Riemann fluxes are dependent on trace values which are obtained numerically. Riemann fluxes computed this way may deviate from the actual physical values. This might explain why many (approximate) Riemann solvers capable of resolving contact discontinuities as well as shocks often exhibit some pathological phenomena among which the so-called carbuncle problem [10, 21, 24] is the most notorious one. Instead of finding the "cure" for such phenomena, some researchers resort to the so-called Riemann-solver-free approaches to avoid such phenomena. Examples include the space-time Conservation Element and Solution Element (CE/SE) method [3, 5], the Nessyahu-Tadmor (NT) scheme [19] and its improved variant [12, 15]. Indeed, these schemes are often referred to as central schemes in contrast to the Riemann-solver-based upwind ones. These central schemes are free of the carbuncle problem and produce entropy-satisfying solutions. In addition, since no Riemann solvers are involved in these central schemes, computation of numerical fluxes do not need the eigen-structure information of the system, which is attractive in governing equations where the eigen structure is not explicitly or easily known.

Numerical methods are required to be of high-resolution in both space and time such that the complex and possibly transient physical features in the simulated flow field are not overly smeared out during the long-time simulation. High-resolution means that the numerical dissipative error and dispersive error inherent to the scheme are small compared to the corresponding physical ones. Traditionally, the second-order accurate schemes were considered as high-resolution schemes. This was true when compared to the dissipative first order methods. Second order schemes have been extensively adopted in many commercial packages due to their simpleness and acceptable accuracy for many engineering applications. However, there are also many applications where second order methods are not adequate. For example, in the field of aeroacoustics, small disturbances (acoustics) often co-exist with strong discontinuities (e.g., shock waves), second order schemes tend to smear out the small disturbances while capturing strong shocks. Other examples include the wake of rotor blades and the flow field around flapping wings of Micro Aerial Vehicles (MAVs) where the flow is highly unsteady and vortex abundant. In this situation, numerical schemes of higher resolution are required to capture the transient vortices.

High resolution methods usually employ high-order (higher than second order) spatial and temporal discretization. Various high-order methods such as the WENO (weighted essentially nonoscillatory) [14] scheme, the discontinuous Galerkin (DG) finite element method [7], the spectral element method (SEM) [22] and the spectral volume method (SVM) [31] have been proposed in the literature and studied by many researchers. High resolution methods may not be of high order in terms of truncation error. An example is the aforementioned CE/SE method [3, 5] that yields high-resolution solutions though it is designed to be second order accurate in both space and time. Solvers based on the CE/SE method [28] preserves the solution profile after long-time advection which otherwise would be overly smeared using the second-order TVD schemes.

In semi-discrete schemes, the high-resolution in the spatial domain and the time domain is achieved separately. Most semi-discrete high-resolution methods focus on the construction of high-order accuracy in space only. Finite difference discretization, e.g., the Backward Difference Formula (BDF), or the multi-step Runge-Kutta method is commonly adopted for the time integration in a semi-discrete scheme. By contrast, in space-time methods, the temporal accuracy is obtained in the same manner as the spatial accuracy. In other words, the time dimension is treated as the $(n+1)$th dimension in addition to the n spatial dimensions. This unified approach offers an alternative way to achieve high-order temporal accuracy.

Besides, space-time methods are very suitable for problems involving moving or deforming domains. Space-time formulation automatically satisfies the so-called Geometric Conservation Law (GCL) [13] as contrasted by the widely used Arbitrary Lagrangian-Eulerian (ALE) methods which do not automatically satisfy the GCL.

Based on the discussions above, space-time methods are attractive since they allow numerical schemes to be Riemann-solver-free, convenient to attain high-order temporal accuracy, and suitable for moving boundary problems.

In addition to the accuracy requirement, high resolution methods must also be able to handle complex geometries where unstructured meshes are usually used in the domain discretization and have high parallelizability for large-scale simulations. The former requires that high-resolution schemes be designed for unstructured meshes from the start. The latter requires that high-resolution schemes be highly compact. The compactness means that only information at the immediate neighboring nodes is required to update the solution at the node under consideration.

The above considerations motivate the authors to design a new scheme that possesses the following desired features:

- space-time formulation;
- high-order accurate in both space and time;
- Riemann-solver-free;
- highly compact;
- suitable for arbitrarily unstructured meshes;
- simple handling of moving and nonreflective boundary conditions.

Indeed, the new scheme presented in this paper were inspired by two of the aforementioned high-resolution methods, the space-time CE/SE method and the space-time DG method.

The DG finite element methods [1,6,7,9] use local discontinuous finite element basis functions to achieve high-order accuracy while keeping the local conservation property similar to that of the finite volume method. The DG method can be semi-discrete or based on space-time formulation. The time integration in the semi-discrete DG method typically adopts an explicit multi-step Runge-Kutta algorithm. The reader can refer to [7] and references therein for a comprehensive review of the DG method, especially the so-called RKDG method. In space-time DG methods [17,20,30], the basis functions contain both spatial and temporal components. The biggest advantage of the DG method is that the stencil is the same regardless of the degree of the basis polynomials. This extreme compactness of the DG method makes it trivial to attain high parallelizability. In the DG method, the inter-element upwind fluxes provide the necessary stabilization in the advection dominant problems. In addition, the *hp* refinement can be easily implemented in this method because hanging nodes and varying degrees of basis functions from element to element are allowed in the DG method.

The CE/SE method introduced in the early 90's [3] has many nontraditional features. It introduces the concepts of space-time conservation elements (CEs) and solution elements (SEs). The space-time domain is filled by nonoverlapping CEs. The boundary of each CE is divided into several sections, with each belonging to a unique SE. Linear variations of the solution are assumed within each SE. Therefore, the CE/SE method is, by design, a second order scheme in both space and time. The final formulation is a result of enforcing the space-time flux conservation on a staggered space-time mesh. The CE/SE method is a Riemann solver-free approach. Even though it is designed to be a second-order scheme, its numerical dissipation is extraordinarily small, making it capable of accurately handling both strong shocks and small disturbances (acoustics) simultaneously [16]. The CE/SE method is able to simulate flows with very low Mach number without using any type of preconditioning. Due to its space-time formulation, the treatment of non-reflective boundary conditions becomes very simple. The original CE/SE method behaves similarly to the fully discrete NT scheme [19], i.e., overly dissipative when small Courant number is used. However, remedies are available to make the CE/SE method Courant number insensitive [4]. For a comprehensive overview of the CE/SE method, the reader is referred to a more recent paper [2] of Chang, the inventor of the method, and the abundant references therein.

In [28], the authors extended the CE/SE method by introducing a universal definition of solution elements (SEs) and conservation elements (CEs) which is independent of the underlying mesh. The staggered space-time mesh, which is the core idea in the CE/SE method, can be constructed for arbitrarily unstructured spatial meshes using the same definitions of SEs and CEs. The scheme in [28] was termed as the Cell Vertex Scheme (CVS) since it updates the solution at vertices and cell centers alternately within each physical time step. Similar to the CE/SE method, the CVS is Riemann-solver-free and second order accurate in both space and time.

The new scheme described in this paper is aimed to make the CVS in [28] high-order accurate by incorporating high-degree discontinuous basis functions within each solution element. Indeed, the new method is a hybrid of the CVS and the DG method and thus is termed as DG-CVS. The combination results in a high-order method for solving hyperbolic conservation laws. The new method offers benefits for both the space-time

DG method and the space-time CE/SE method. The benefits for the space-time CE/SE method is two fold. First, the new method increases the CE/SE method's temporal and spatial accuracies by employing high-degree polynomial basis functions inside the SE. Second, the new method mitigates the phenomenon that excess dissipation occurs when very small timesteps are used. Actually, as will be demonstrated in the numerical tests, when high-degree (e.g., $p4$) basis functions are employed, the absolute error becomes smaller when smaller time steps are used. The benefit for the space-time DG method is the elimination of the need of explicitly upwinded flux functions (i.e., Riemann solvers). As pointed out in [28], the upwind direction is automatically detected if the discretization is based on the space-time flux conservation on staggered space-time CEs.

This paper aims to present the detailed formulation and some implementation issues and demonstrate the performance of the new method for the hyperbolic conservation laws. Parts of this paper have been presented in the author's three other recent AIAA conference papers [26,27,29].

The rest of the paper is organized as follows. Section 2 provides a detailed description of the basic formulation of the new method for the 1-D linear scalar advection equation. Section 3 discusses the choices of basis functions. Section 4 provides a quadrature-free method to integrate within general polygons and polygonal cylinders. Section 5 explains how to handle some common boundary conditions in the present DG-CVS framework. Several test cases are provided in Section 6 to demonstrate the performance of the present scheme for hyperbolic conservation laws including scalar advection equations and compressible Euler equations. Finally, concluding remarks are given in Section 7.

2 The high-order cell vertex scheme

In this section, a detailed description of the new high-order cell-vertex scheme for 1-D scalar advection problem is presented. Subsections 2.1 and 2.2 review the alternate cell-vertex solution updating strategy and the universal definitions of conservation elements and solution elements which have been presented in [28]. Section 2.3 provides the basic formulation of the present new DG-CVS for one dimension cases.

2.1 Alternate cell-vertex solution updating strategy

The so-called cell-vertex scheme (CVS) introduced in [28] is an extension of the CE/SE method. The CVS inherits the core idea of the CE/SE method using a staggered space-time mesh to enforce the space-time flux conservation. However, the staggered space-time mesh is realized through the alternate solution updating at cell centroids and vertices within a physical time step, which conceptually departs from the original CE/SE method. In CVS, unknowns are stored at both vertices and cell centroids of the spatial mesh. However, the solutions at vertices and cell centroids are updated at different time levels within each time step in an alternate fashion. At the beginning of each physical time step, the solution is assumed known at the vertices of the mesh, either given as the initial condition or obtained from the previous time step. Inside each new time step, the solution is updated in two successive steps. The first step updates the solution at cell centroids at the half-time level ($t^{n+1/2}$) based on the known vertex solutions at the previous time level (t^n). The second step updates the solution at vertices at the new time level (t^{n+1}) based on the known cell solutions at the previous half-time level ($t^{n+1/2}$). The same process is repeated for new time steps.

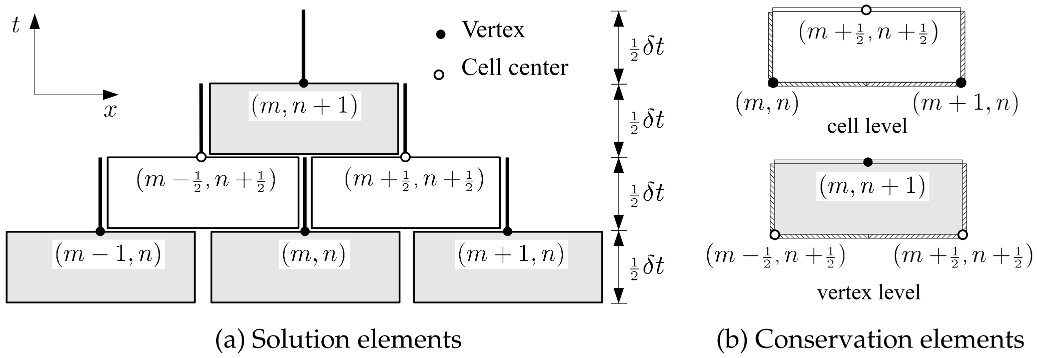
Since the CVS is a derivative of the CE/SE method, it inherits the advantageous features of the latter. The most important feature is that CVS, like the CE/SE method, is a Riemann-solver-free approach. In addition, the CVS has its own merit, namely, it allows universal definitions of the conservation element (CE) and the solution element (SE) which is applicable to arbitrarily unstructured spatial meshes. The next subsection will review such definitions.

2.2 Universal definitions of solution element and conservation element

Here, the 1-D spatial mesh is taken as the example to illustrate the universal definitions of CEs and SEs. Fig. 1(a) shows the neighboring SEs on $x-t$ space-time domain that is based on 1-D spatial meshes. Here the indices in (m,n) are used to denote space-time coordinates of the space-time node, respectively. The space-time *node* can be at the cell level or the vertex level. The SE associated with the space-time node (m,n) is defined as the rectangular region plus the vertical volumeless segment above the rectangle. Note that the height of each rectangle is $\delta t/2$, namely, half the time step. Within each SE, the solution satisfies a polynomial function of certain degree.

Fig. 1(b) shows the corresponding space-time conservation elements (CEs) at the cell level and the vertex level, respectively. The boundary of each CE is composed of several sections with each section belonging to a unique SE defined in Fig. 1(a). For example, when enforcing the space-time flux conservation in the CE associated with the cell centroid $(m+1/2,n+1/2)$, the space-time fluxes on the left side boundary and left bottom boundary of the CE are evaluated using the polynomial function inside the SE associated with vertex (m,n), the space-time fluxes on the right side boundary and the right bottom boundary of the CE are evaluated using the polynomial function inside the SE associated with vertex $(m+1,n)$, and the space-time fluxes on the top boundary of the CE are evaluated using the polynomial function inside the SE associated with the cell itself (which is to be determined). The CE boundaries in Fig. 1(b) are indicated by distinctive thatch patterns for sections belonging to different SEs. The same space-time flux conservation can be enforced similarly in the vertex-level CE, as shown in Fig. 1(b) (bottom).

Remark 2.1. As can be seen in Fig. 1, the present staggered space-time mesh is layered (structured in time) which is in contrast to the space-time DG method in [20] where the space-time mesh can be layered or fully unstructured. The geometric shape of the CE and the SE is essentially the same except that the SE has an extra *volumeless* vertical segment which is also the side boundary of the CE at the next level.



(a) Solution elements (b) Conservation elements

Figure 1: Solution elements (SEs) and conservation elements (CEs) in the $x-t$ domain.

The resultant SEs and CEs based on the definitions described above are the same as those in the original CE/SE method on 1-D spatial meshes. However, deviation occurs in higher spatial dimensions. The same definitions can be easily extended to higher-dimensional arbitrary meshes (cf. Fig. 2) without any ambiguity. The advantage of the present CE and SE definitions is that the staggered space-time mesh can be constructed using the existing mesh cell-vertex topology information. It can be concluded that this novel definitions of CEs and SEs are independent of the underlying mesh. Of course, the geometric shape of the resulting CE does depends on the underlying spatial meshes. Due to this reason, the definition of SEs and CEs introduced in [28] is considered as *universal*.

2.3 Basic CVS formulation

To illustrate the derivation of the basic CVS formulation, we consider the following one-dimensional scalar advection problem with appropriate initial and boundary conditions

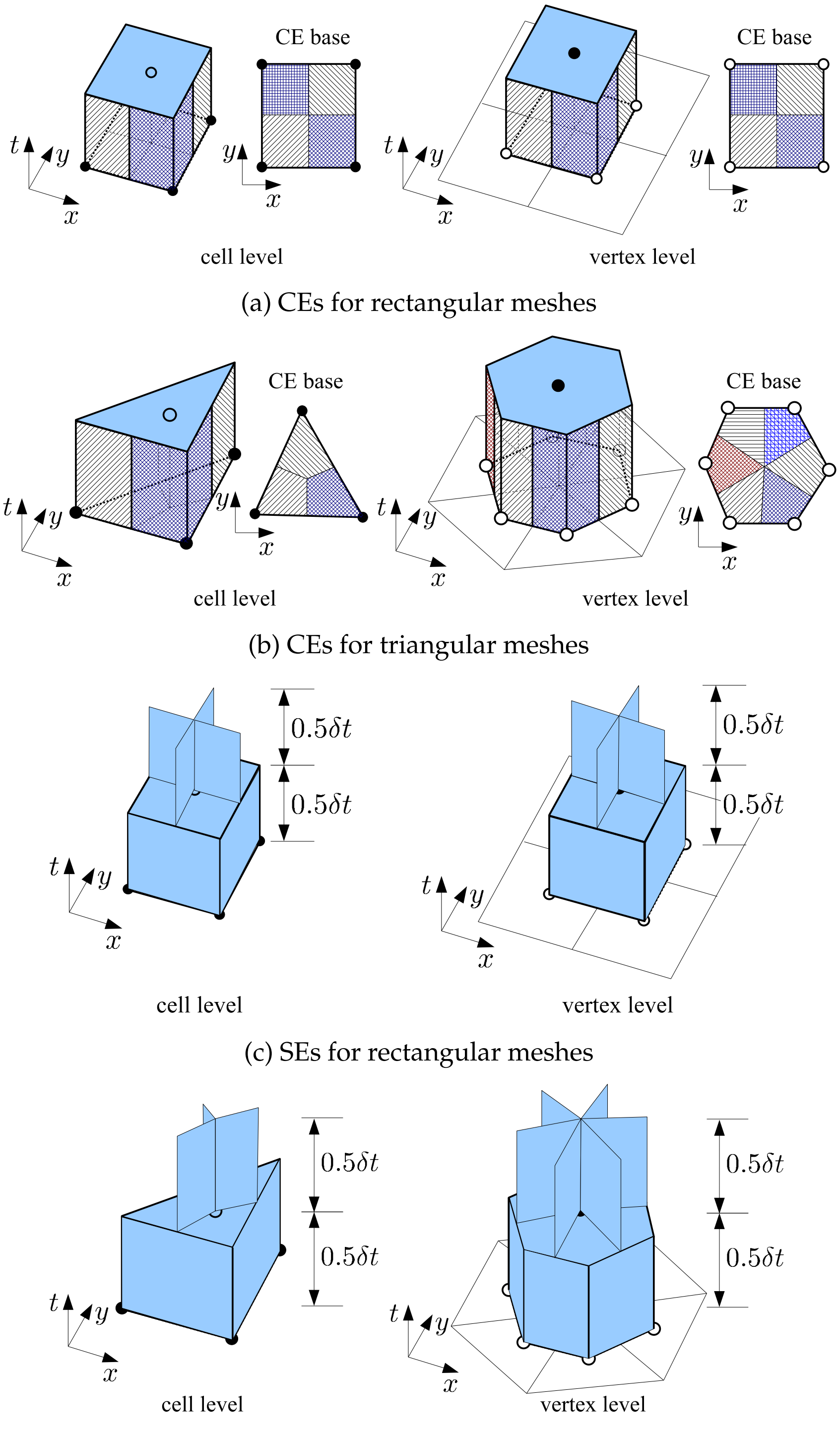
$$\frac{\partial u(x,t)}{\partial t}+\frac{\partial f(u)}{\partial x}=0, \tag{2.1}$$

where u is the advected quantity and f is the flux.

Following the idea of DG, an approximate solution u^h is sought within each space-time solution element (SE), denoted as K. When restricted to the SE, u^h belongs to the finite dimensional space $\mathcal{U}(K)$ such that

$$u^h(x,t)=\sum_{j=1}^{N}\phi_j s_j, \tag{2.2}$$

where $\{\phi_j\}_{j=1}^N$ are some type of polynomial basis functions, $\{s_j\}_{j=1}^N$ are the unknowns to be determined and N is the number of basis functions depending on the degree of the polynomial function.



(a) CEs for rectangular meshes

(b) CEs for triangular meshes

(c) SEs for rectangular meshes

(d) SEs for triangular meshes

Figure 2: Conservation elements (CEs) and solution elements (SEs) in the x-y-t domain.

Galerkin orthogonality states that for all test functions $v^h \in \mathcal{U}(K)$,

$$\int_\Omega v^h \Big(\frac{\partial u^h}{\partial t} + \frac{\partial f^h}{\partial x} \Big) d\Omega = 0, \tag{2.3}$$

where Ω is the conservation element (CE) corresponding to the solution element K. Note that the difference between CEs and SEs is the *volumeless* vertical segment as shown in Fig. 1(a).

It is sufficient to replace v^h in Eq. (2.3) by each of the basis functions $\{\phi_j\}_{j=1}^N$. Integrating the resulting weak form by parts yields

$$\int_\Omega \Big[\frac{\partial \phi_i}{\partial t} u^h + \frac{\partial \phi_i}{\partial x} f^h \Big] d\Omega = \int_\Gamma \phi_i F^h_{\mathbf{n}} d\Gamma, \tag{2.4}$$

where

$$F^h_{\mathbf{n}} = F^h \cdot \mathbf{n} = \big(f^h, u^h\big) \cdot (n_x, n_t) \tag{2.5}$$

is the space-time flux normal to the boundary of the space-time CE, $\mathbf{n} = (n_x, n_t)$ is the outward unit normal of the CE boundary, $\Gamma = \partial\Omega$ is the boundary of the CE under consideration.

The cell level CE shown in Fig. 1(b) is taken as a specific example here. As shown in Fig. 3, divide Γ into five sections Γ_1, Γ_2, Γ_3, Γ_4 and Γ_5 where Γ_1 belongs to the SE associated with $(m+1/2, n+1/2)$ where the solution is being sought, Γ_2 and Γ_3 the SE associated with (m,n) and Γ_4 and Γ_5 the SE associated with $(m+1,n)$. Note that the solutions at nodes (m,n) and $(m+1,n)$ are known since they are at the previous time level. As a result, (2.4) becomes

$$\begin{aligned} &\int_\Omega \Big[\frac{\partial \phi_i}{\partial t} u^h_{m+\frac{1}{2},n+\frac{1}{2}} + \frac{\partial \phi_i}{\partial x} f^h_{m+\frac{1}{2},n+\frac{1}{2}} \Big] d\Omega - \int_{\Gamma_1} \phi_i F^h_{\mathbf{n}(m+\frac{1}{2},n+\frac{1}{2})} d\Gamma \\ &= \int_{\Gamma_2+\Gamma_3} \phi_i F^h_{\mathbf{n}(m,n)} d\Gamma + \int_{\Gamma_4+\Gamma_5} \phi_i F^h_{\mathbf{n}(m+1,n)} d\Gamma, \end{aligned} \tag{2.6}$$

where the left hand side contains the unknowns at the time level $t^{n+1/2}$ and the right hand side contains the known solution at the time level t^n. In (2.6), the subscripts represent the space-time indices in the space-time domain. On boundary Γ_1, the outward unit normal $\mathbf{n} = (0,1)$. Therefore,

$$F^h_{\mathbf{n}(m+\frac{1}{2},n+\frac{1}{2})} = u^h_{m+\frac{1}{2},n+\frac{1}{2}} \qquad \text{on } \Gamma_1.$$

Similarly, on boundaries Γ_3 and Γ_4, the outward unit normal $\mathbf{n} = (0,-1)$, leading to

$$\begin{aligned} F^h_{\mathbf{n}(m,n)} &= -u^h_{m,n} && \text{on } \Gamma_3, \\ F^h_{\mathbf{n}(m+1,n)} &= -u^h_{m+1,n} && \text{on } \Gamma_4. \end{aligned}$$

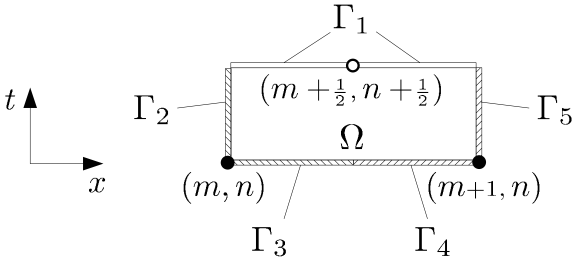


Figure 3: Illustration of space-time flux conservation in a conservation element at the cell level.

Substituting these into (2.6) to obtain

$$\begin{aligned}&\int_\Omega\Big[\frac{\partial\phi_i}{\partial t}u^h_{m+\frac{1}{2},n+\frac{1}{2}}+\frac{\partial\phi_i}{\partial x}f^h_{m+\frac{1}{2},n+\frac{1}{2}}\Big]d\Omega-\int_{\Gamma_1}\phi_i u^h_{m+\frac{1}{2},n+\frac{1}{2}}d\Gamma\\&=\int_{\Gamma_2}\phi_i F^h_{\mathbf{n}(m,n)}d\Gamma-\int_{\Gamma_3}\phi_i u^h_{m,n}d\Gamma-\int_{\Gamma_4}\phi_i u^h_{m+1,n}d\Gamma+\int_{\Gamma_5}\phi_i F^h_{\mathbf{n}(m+1,n)}d\Gamma.\end{aligned}\tag{2.7}$$

Note that even though the side boundaries of the stationary CE shown in Fig. 3 have outward unit normals $\mathbf{n}=(\pm 1,0)$ which can be used to simplify the fluxes across Γ_2 and Γ_5, for general moving meshes the side boundaries are not vertical. For this reason, the flux terms for Γ_2 and Γ_5 are kept in their general forms without loss of generality. In (2.7),

$$F^h_{\mathbf{n}(m,n)}=\big(f(u^h_{m,n}),u^h_{m,n}\big)\cdot\mathbf{n}_{\Gamma_2}$$

and $u^h_{m,n}$ is computed according to the known solution $\mathbf{s}_{m,n}$ at the space-time node (m,n) using (2.2). $f(u^h_{m,n})$ is obtained from $u^h_{m,n}$. $F^h_{\mathbf{n}(m+1,n)}$ can be evaluated similarly. Therefore, the computation of the flux terms in (2.7) does not involve any Riemann solver.

Eq. (2.7) is for updating the solution at the space-time node $(m+1/2,n+1/2)$, which is at the cell level. For vertex level nodes, similar formulation can be derived. Note that (2.7) is the most general formulation for 1-D scalar advection problems. The formulation is based on the universal definitions of CEs and SEs described in Subsection 2.2. Thus it is readily extensible for higher spatial dimensions.

2.4 Linear flux case

The flux f in (2.7) can be a linear or nonlinear function of the advected quantity depending on the specific problem. When the flux f is a linear function of u, e.g., $f=au$ where a is the constant advection speed, a linear equation system can be obtained from (2.7). Substituting (2.2) into (2.7), one gets

$$\begin{aligned}&\int_\Omega\Big[\frac{\partial\phi_i}{\partial t}\Big(\sum_{j=1}^N\phi_j s_{j(m+\frac{1}{2},n+\frac{1}{2})}\Big)+a\frac{\partial\phi_i}{\partial x}\Big(\sum_{j=1}^N\phi_j s_{j(m+\frac{1}{2},n+\frac{1}{2})}\Big)\Big]d\Omega-\int_{\Gamma_1}\phi_i\Big(\sum_{j=1}^N\phi_j s_{j(m+\frac{1}{2},n+\frac{1}{2})}\Big)d\Gamma\\&=\int_{\Gamma_2}\phi_i F^h_{\mathbf{n}(m,n)}d\Gamma-\int_{\Gamma_3}\phi_i u^h_{m,n}d\Gamma-\int_{\Gamma_4}\phi_i u^h_{m+1,n}d\Gamma+\int_{\Gamma_5}\phi_i F^h_{\mathbf{n}(m+1,n)}d\Gamma,\end{aligned}\tag{2.8}$$

which can be expressed in matrix-vector form as

$$\mathbf{M}\mathbf{s}=\mathbf{b}, \tag{2.9}$$

where $\mathbf{M}$ is a non-symmetric $N\times N$ matrix, $\mathbf{s}$ is an $N\times 1$ vector containing the unknowns, s_j, and $\mathbf{b}$ is another $N\times 1$ vector containing known components. The elements of $\mathbf{M}$ and $\mathbf{b}$, M_{ij} and b_i are explicitly given as

$$M_{ij}=\int_\Omega \Big[\frac{\partial\phi_i}{\partial t}\phi_j+\Big(\frac{\partial\phi_i}{\partial x}\phi_j\Big)a\Big]d\Omega-\int_{\Gamma_1}\phi_i\phi_j d\Gamma, \tag{2.10a}$$

$$b_i=\int_{\Gamma_2}\phi_i F^h_{\mathbf{n}(m,n)}d\Gamma-\int_{\Gamma_3}\phi_i u^h_{m,n}d\Gamma-\int_{\Gamma_4}\phi_i u^h_{m+1,n}d\Gamma+\int_{\Gamma_5}\phi_i F^h_{\mathbf{n}(m+1,n)}d\Gamma. \tag{2.10b}$$

As can be seen, a small linear equation system (whose size depends on the number of basis functions employed) must be solved for each space-time node. For this reason, the current CVS can be considered as a point implicit scheme.

2.5 Nonlinear flux case

When the flux f is a nonlinear function of u, Eq. (2.7) results in a set of nonlinear equations. The standard Newton-Raphson method can be used to solve the nonlinear equation set for $\mathbf{s}$. First denote for $i=1,2,\cdots,N$,

$$\begin{aligned}G_i(\mathbf{s})=&\int_\Omega\Big[\frac{\partial\phi_i}{\partial t}u^h_{m+\frac12,n+\frac12}+\frac{\partial\phi_i}{\partial x}f^h_{m+\frac12,n+\frac12}\Big]d\Omega-\int_{\Gamma_1}\phi_i u^h_{m+\frac12,n+\frac12}d\Gamma\\&-\int_{\Gamma_2}\phi_i F^h_{\mathbf{n}(m,n)}d\Gamma+\int_{\Gamma_3}\phi_i u^h_{(m,n)}d\Gamma+\int_{\Gamma_4}\phi_i u^h_{(m+1,n)}d\Gamma\\&-\int_{\Gamma_5}\phi_i F^h_{\mathbf{n}(m+1,n)}d\Gamma=0,\end{aligned} \tag{2.11}$$

which is obtained by moving all terms of (2.7) to the left hand side.

Newton-Raphson method states that the solution can be updated iteratively by means of

$$\mathbf{s}^{(\tau+1)}=\mathbf{s}^{(\tau)}-\mathbf{J}^{-1}\mathbf{G}(\mathbf{s}^{(\tau)}), \tag{2.12}$$

where the superscript τ represents the iteration step. The Jacobian matrix $\mathbf{J}=\partial\mathbf{G}/\partial\mathbf{s}$ depending on the specific functional form of the flux f. For instance, for the inviscid Burgers equation,

$$f=\frac12 u^2,$$

namely,

$$f^h=\frac12(u^h)^2=\frac12\Big[\sum_{j=1}^N(\phi_j s_j)\Big]^2,$$

whose derivatives with respect to s_j are

$$\frac{\partial f^h}{\partial s_j} = \phi_j \sum_{j=1}^{N} (\phi_j s_j) = \phi_j u^h. \tag{2.13}$$

The derivatives of u^h with respect to s_j are

$$\frac{\partial u^h}{\partial s_j} = \phi_j. \tag{2.14}$$

Eqs. (2.13) and (2.14) together with (2.11) can be used to obtain the Jacobian matrix. One observes that the Jacobian matrix has the same form of the matrix whose entries are given in (2.10a) except that a in (2.10a) is replaced by $u^{(\tau)}$ which is u^h at the previous iteration step, namely, the entries of $\mathbf{J}$ are given by

$$J_{ij} = \int_{\Omega} \left[\frac{\partial \phi_i}{\partial t} \phi_j + \left(\frac{\partial \phi_i}{\partial x} \phi_j \right) u^{(\tau)} \right] d\Omega - \int_{\Gamma_1} \phi_i \phi_j d\Gamma. \tag{2.15}$$

To update $\mathbf{s}$ using (2.12), one can first compute the solution increment $\delta\mathbf{s}$ by solving

$$\mathbf{J}\delta\mathbf{s} = -\mathbf{G}(\mathbf{s}^{(\tau)}),$$

which is a linear equation system and can be solved by a direct or iterative method. The solution is then updated via

$$\mathbf{s}^{(\tau+1)} = \mathbf{s}^{(\tau)} + \delta\mathbf{s}.$$

The initial guess of the solution, $\mathbf{s}^{(0)}$, is currently taken as the average of the solutions at the surrounding nodes at the previous time level. Numerical tests in Sections 6.3.2 and 6.3.4 indicate that 2-5 iterations are sufficient in the case of the inviscid Burgers equation to reach convergence.

Remark 2.2. A distinction from the DG or space-time DG methods is that no Riemann-solver-typed flux functions are employed in the present method when evaluating the space-time fluxes across the CE side boundaries. The apparent lack of upwinded fluxes does not cause numerical instability. Actually, as pointed out in [28], the upwind direction is automatically detected when using the staggered space-time formulation.

Remark 2.3. Compared to the original CVS in [28] which is only second order accurate, the present DG-CVS can be arbitrarily high order accurate, at least in theory, thus achieving improvements over the original CVS or the CE/SE method. Note that in the original CVS [28] as well as the CE/SE method [3], the time derivative of the solution is obtained from the spatial derivative of the solution using the original governing equation (2.1). By contrast, in the new DG-CVS, the time derivative of the solution is treated as an independent unknown and handled in the same way as that for the spatial derivative of the solution. Therefore, even for the same second-order version, the new DG-CVS deviates from the original CVS or the CE/SE method.

Remark 2.4. The derivation of the new scheme is based on the same highly compact stencil regardless of the order as in the DG method. In other words, the new solution at a space-time node relies only on solutions at the immediate neighboring nodes at the previous time level regardless of the degree of basis polynomials. This feature is desirable for two reasons: (i) the solution at the new time level relies on a finite region of dependence according to the theory of characteristics, and (ii) compact schemes are highly parallelizable on distributed memory computer platforms.

3 Choice of basis functions

The solution inside each solution element is assumed to satisfy a polynomial function of certain degree (cf. Eq. (2.2)). Different forms of polynomials can be used. For example, Lagrange polynomials, Taylor polynomials, Chebyshev polynomials and Legendre polynomials are all possible choices. Choice of polynomial functions will affect the accuracy and efficiency of the solution procedure.

In the present implementation, the Taylor polynomials is chosen. The bivariate Taylor polynomial of degree p in 1D can be expressed as

$$\begin{aligned} u^h(x,t) &= \sum_{r=0}^{p} \frac{1}{r!}\Big(\Delta x\frac{\partial}{\partial x}+\Delta t\frac{\partial}{\partial t}\Big)^r u(x_0,t_0) \\ &= \sum_{r=0}^{p}\Big[\frac{1}{r!}\sum_{i=0}^{r}\begin{pmatrix} r \\ i \end{pmatrix}\Delta x^{r-i}\Delta t^i\Big(\frac{\partial^r u}{\partial x^{r-i}\partial t^i}\Big)_{(x_0,t_0)}\Big], \end{aligned} \tag{3.1}$$

where (x_0,t_0) is the space-time vertex-level or cell-level node whose solutions are to be determined. p is the degree of the polynomial. $\Delta x = x - x_0$ and $\Delta t = t - t_0$ are the spatial and temporal distances to (x_0,t_0), respectively. The Taylor polynomial (3.1) is an approximation of the solution in the vicinity of the space-time node (x_0,t_0) using the derivative information of the solution at that node.

Comparing (3.1) and (2.2), one can extract the basis functions ϕ_j's and the unknowns s_j's. The number of the basis functions and unknowns is

$$N = \frac{1}{2}(p+1)(p+2).$$

Take the quadratic case ($p=2$ and $N=6$) as an example. One has the basis functions

$$\phi_1 = 1, \quad \phi_2 = \Delta x, \quad \phi_3 = \frac{1}{2}\Delta x^2, \quad \phi_4 = \Delta t, \quad \phi_5 = \Delta x\Delta t, \quad \phi_6 = \frac{1}{2}\Delta t^2,$$

which are the functions of Δx and Δt, and the solution vector components

$$\begin{aligned} s_1 &= u_0^h, & s_2 &= \Big(\frac{\partial u^h}{\partial x}\Big)_0, & s_3 &= \Big(\frac{\partial^2 u^h}{\partial x^2}\Big)_0, \\ s_4 &= \Big(\frac{\partial u^h}{\partial t}\Big)_0, & s_5 &= \Big(\frac{\partial^2 u^h}{\partial x\partial t}\Big)_0, & s_6 &= \Big(\frac{\partial^2 u^h}{\partial t^2}\Big)_0, \end{aligned}$$

which are the unknown itself and its spatial and temporal derivatives of various order.

For 2D problems, the following trivariate Taylor polynomials are used:

$$\begin{aligned} u^h(x,y,t) &= \sum_{r=0}^{p} \frac{1}{r!} \Big(\Delta x \frac{\partial}{\partial x} + \Delta y \frac{\partial}{\partial y} + \Delta t \frac{\partial}{\partial t} \Big)^r u(x_0,y_0,t_0) \\ &= \sum_{r=0}^{p} \Big[\frac{1}{r!} \sum_{i=0}^{r} \sum_{j=0}^{i} \left(\begin{array}{c} r \\ i \end{array} \right) \left(\begin{array}{c} i \\ j \end{array} \right) \Delta x^{r-i} \Delta y^{i-j} \Delta t^j \Big(\frac{\partial^r u}{\partial x^{r-i} \partial y^{i-j} \partial t^j} \Big)_{(x_0,y_0,t_0)} \Big], \end{aligned} \tag{3.2}$$

where

$$\left(\begin{array}{c} r \\ i \end{array} \right) \left(\begin{array}{c} i \\ j \end{array} \right) = \frac{r!}{(r-i)!(i-j)!j!}$$

is the trinomial coefficient. Obviously, the number of basis functions equals

$$\frac{1}{3!}(p+1)(p+2)(p+3).$$

Since typical integrals involves double or triple products between polynomials, it is essential to normalize Δx, Δy, and Δt in the basis functions using the local cell size for better accuracy, which is especially true when polynomials of high degrees are used.

The Taylor polynomial is chosen for three reasons. First, the Taylor polynomial has no restrictions on the geometric shape of the solution element (SE). The SE can be of arbitrary shape which is typically polygonal cylinder on spatially unstructured meshes (cf. Fig. 2(b)). By contrast, Lagrange polynomials are only well defined on simplicial elements or elements allowing tensor products, and Chebyshev polynomials and Legendre polynomials are only well defined on elements allowing tensor products.

Second, with Taylor polynomials, the basis functions are polynomials with respect to the physical coordinates Δx and Δt. Other polynomials usually define the basis function using reference coordinates (e.g., ξ and η). The numerical integration in Eq. (2.8) involves the derivatives of the basis function with respect to the physical coordinates. With Taylor polynomials, the derivatives can be directly obtained by taking the derivative of the basis functions with respect to Δx and Δt without resorting to the chain rule as is required when other polynomials are employed.

Third, the derivatives of Taylor polynomials are a lower order subset of the original Taylor polynomials. This allows efficient implementation when evaluating integrals involving products of polynomials. For example, the quadratic 2-D basis functions are tabulated in Table 1.

Remark 3.1. The above discussions of the advantages of Taylor polynomials over other polynomials are completely based on the implementation point of view. One should keep in mind that high degree Taylor polynomials are notorious in generating severely ill-conditioned systems. It is well known that the number of reliable digits of the solution of linear systems is related to the conditioning of the system. As a result, high degree Taylor polynomials are expected to yield inaccurate solutions due to the ill conditioning

Table 1: Derivatives of quadratic 2-D basis functions.

i	unknowns (s_i)	ϕ_i	$\frac{\partial\phi_i}{\partial x}$	$\frac{\partial\phi_i}{\partial y}$	$\frac{\partial\phi_i}{\partial t}$
1	u_0^h	1	0	0	0
2	$\left(u_x^h\right)_0$	Δx	1	0	0
3	$\left(u_y^h\right)_0$	Δy	0	1	0
4	$\left(u_{xx}^h\right)_0$	$\frac{1}{2}\Delta x^2$	Δx	0	0
5	$\left(u_{xy}^h\right)_0$	$\Delta x\Delta y$	Δy	Δx	0
6	$\left(u_{yy}^h\right)_0$	$\frac{1}{2}\Delta y^2$	0	Δy	0
7	$\left(u_t^h\right)_0$	Δt	0	0	1
8	$\left(u_{xt}^h\right)_0$	$\Delta x\Delta t$	Δt	0	Δx
9	$\left(u_{yt}^h\right)_0$	$\Delta y\Delta t$	0	Δt	Δy
10	$\left(u_{tt}^h\right)_0$	$\frac{1}{2}\Delta t^2$	0	0	Δt

of the system, which will be verified in the section of numerical tests. Therefore, choosing Taylor polynomials is not wise from the accuracy point of view. However, the purpose of this paper is to demonstrate the efficacy of the DG-CVS idea, so Taylor polynomials are employed for quick implementation. Implementing other basis polynomials remains our future work.

4 Integration within general polygons and polygonal cylinders

As can be seen from Eqs. (2.10a) and (2.10b) together with Fig. 3, the space-time CVS involves both the volume integral and surface integrals. These integrals must be integrated either numerically or analytically. Obviously, for 1D meshes (Fig. 1(b) and 2D rectangular meshes (Fig. 2(a)), we can directly apply the Gaussian quadrature rule to compute all integrals. However, if the underlying spatial mesh is unstructured (Fig. 2(b)), the vertex-level CEs are general polygonal cylinders containing general polygonal bases and quadrilateral side faces where the Gaussian quadrature rule cannot be directly applied.

We need to evaluate the following integrals to compute the right hand side vector (Eq. (2.10b)) of the linear system (2.9):

• *surface integrals involving the side faces of the CE.* The side faces of CEs are always quadrilaterals (for 2D meshes). Actually, for non-moving meshes, they are rectangles. Therefore, Gaussian quadrature rule can be applied to numerically integrate the surface integrals on these faces.

• *surface integrals involving the bottom face of the CE.* The bottom face of the CE is a general polygon. For example, at the cell level, the bottom face of the CE is a triangle and the Gaussian quadrature rule can be used. By contrast, at the vertex level, the bottom face of the CE is a general

n-sided polygon. The polygonal shape of faces does not incur difficulties in the integration since the bottom face is divided into several quadrilaterals with each belonging to a solution element at the previous time level. Therefore, Gaussian quadrature rule can be applied here too.

Therefore, the Gaussian quadrature rule can be used to evaluate all integrals in the computation of the right hand side vector.

To compute the left hand side matrix (Eq. (2.10a)) of the linear system, we need to evaluate the following integrals:

- *surface integrals involving the top face of the CE*. The top face of the CE is identical to the bottom face of the CE for stationary spatial meshes. However, since the entire top face belongs to the solution element whose solutions are being sought, the n-sided polygon is treated as a whole. The Gaussian quadrature rule cannot be directly applied here.

- *volume integral involving the volume of the CE*. The volume is a general n-sided polygonal cylinder where the Gaussian quadrature rule cannot be directly applied.

Therefore, for those integrals in the computation of the left hand side matrix, the Gaussian quadrature rule cannot be directly applied. This section addresses the procedure to circumvent this difficulty.

4.1 Converting volume integrals to surface integrals

The integrand in the volume integral (cf. Eq. (2.10a)) has the following general form

$$q(x,y,t)=\Delta x^l \Delta y^m \Delta t^n, \tag{4.1}$$

which results from the products between Taylor polynomials. In (4.1), $l,m,$ and n are integer exponents. For notational simplicity, denote

$$\tilde{x}=\Delta x=x-x_0, \quad \tilde{y}=\Delta y=y-y_0 \quad \text{and} \quad \tilde{t}=\Delta t=t-t_0.$$

Then (4.1) becomes

$$q(x,y,t)=\tilde{x}^l\tilde{y}^m\tilde{t}^n. \tag{4.2}$$

Computing volume integrals with integrands of the form as in (4.2) is equivalent to computing the moments of arbitrary order of polyhedra. In the current case, the polyhedra are polygonal cylinders. The method of using the divergence theorem explained in [8] is adopted to reduce the volume integral to surface integrals and further to line integrals.

To evaluate a scalar integral inside a space-time volume Ω, i.e., $\int_\Omega q d\Omega$, we first construct a vector function

$$\mathbf{Q}(x,y,t)=Q_1\mathbf{i}_1+Q_2\mathbf{i}_2+Q_3\mathbf{i}_3,$$

where $\mathbf{i}_1,\mathbf{i}_2$ and $\mathbf{i}_3$ denote the unit vectors along the x, y, and t-directions, respectively, such that

$$q=\nabla\cdot\mathbf{Q}. \tag{4.3}$$

With the aid of the auxiliary function **Q**, one can convert the volume integral into a surface integral via the divergence theorem as follows:

$$\int_\Omega q(x,y,t)d\Omega = \int_\Omega \nabla\cdot\mathbf{Q}d\Omega = \int_{\partial\Omega}\mathbf{Q}\cdot\mathbf{n}d\Gamma. \tag{4.4}$$

Without loss of generality, we can define the following for **Q**

$$Q_1=0, \quad Q_2=0 \quad \text{and} \quad Q_3(x,y,t)=\int q(x,y,t)dt+c(x,y) \tag{4.5}$$

to satisfy (4.3). In (4.5), $\int qdt$ is an indefinite integral with respect to t and $c(x,y)$ is an arbitrary function independent of t.

Substituting (4.5) into (4.4) to obtain

$$\int_\Omega q(x,y,t)d\Omega=\int_{\partial\Omega}Q_3 n_t dA=\int_{\partial\Omega}\Big(\int q(x,y,t)dt\Big)n_t dA+\int_{\partial\Omega}c(x,y)n_t dA, \tag{4.6}$$

where n_t is the outward unit normal in the t-direction of the surface. $\int_{\partial\Omega}c(x,y)n_t d\Gamma$ equals zero for enclosed surfaces since $c(x,y)$ is independent of t and results in zero net contributions. Therefore, it suffices to evaluate $\int_\Omega q d\Omega$ via

$$\int_\Omega q(x,y,t)d\Omega=\int_{\partial\Omega}\Big(\int q(x,y,t)dt\Big)n_t dA. \tag{4.7}$$

For stationary meshes, the side surfaces of the polygonal cylinders are vertical to the x-y plane (i.e., $n_t=0$), so only the top and bottom faces of the CE have nonzero n_t (i.e., $n_t=\pm1$). Hence, (4.7) can be computed as

$$\int_\Omega q(x,y,t)d\Omega=\int_{\Gamma_{\text{top}}}\Big(\int q(x,y,t)dt\Big)dA-\int_{\Gamma_{\text{bot}}}\Big(\int q(x,y,t)dt\Big)dA. \tag{4.8}$$

Furthermore, on the top surface, $\tilde{t}=\Delta t=0$, therefore, the first integral in Eq. (4.8) is zero. (4.8) reduces to

$$\int_\Omega q(x,y,t)d\Omega=-\int_{\Gamma_{\text{bot}}}\Big(\int q(x,y,t)dt\Big)dA. \tag{4.9}$$

For simple polynomials, the indefinite integral can be evaluated analytically, i.e.,

$$\int q(x,y,t)dt=\int \tilde{x}^l\tilde{y}^m\tilde{t}^n dt=\frac{1}{n+1}\tilde{x}^l\tilde{y}^m\tilde{t}^{n+1}. \tag{4.10}$$

Since the bottom face of the CE is a horizontal one, $\tilde{t}$ is a constant on the surface. Considering this fact and substituting (4.10) into (4.9) leads to

$$\int_\Omega q(x,y,t)d\Omega=-\frac{1}{n+1}\tilde{t}^{n+1}\int_{\Gamma_{\text{bot}}}\tilde{x}^l\tilde{y}^m dA. \tag{4.11}$$

As can be seen the volume integral $\int_\Omega q(x,y,t)d\Omega$ has been reduced to a surface integral (4.11).

4.2 Converting surface integral to line integrals

The surface integral (4.11) is an integral over a general polygon which is the bottom face of the CE. In addition, the surface integral in (2.10a) involves the top face of the CE. For stationary meshes, the top surface and the bottom surface of the same CE are identical.

One can repeat the procedure in the previous subsection to convert the surface integral into line integrals. Alternatively, one can apply Green's theorem to reach the same goal.

According to Green's Theorem

$$\int_A \tilde{x}^l \tilde{y}^m dA = \oint_{\partial A}\Big(\int \tilde{x}^l \tilde{y}^m dx\Big)dy = \sum_{i=1}^{L}\frac{1}{l+1}\int_{\partial A_i}\tilde{x}^{l+1}\tilde{y}^m dy, \tag{4.12}$$

where L is the number of edges in the polygon.

For each line segment, the two ending points are denoted as $(\hat{x}_1,\hat{y}_1)$ and $(\hat{x}_2,\hat{y}_2)$. We can parametrize $\tilde{x}$ and $\tilde{y}$ according to

$$\hat{x}(\lambda) = \hat{x}_1 + \lambda(\hat{x}_2 - \hat{x}_1), \tag{4.13a}$$

$$\hat{y}(\lambda) = \hat{y}_1 + \lambda(\hat{y}_2 - \hat{y}_1), \tag{4.13b}$$

where $0 \le \lambda \le 1$. Since

$$dy = d\hat{y} = (\hat{y}_2 - \hat{y}_1)d\lambda,$$

the line integral in (4.12) becomes

$$\int_{\partial A_i}\tilde{x}^{l+1}\tilde{y}^m dy = (\hat{y}_2 - \hat{y}_1)\int_0^1 (\hat{x}_1 + \lambda(\hat{x}_2 - \hat{x}_1))^{l+1}(\hat{y}_1 + \lambda(\hat{y}_2 - \hat{y}_1))^m d\lambda. \tag{4.14}$$

4.3 Quadrature-free integration of line integrals

The line integral in (4.14) has the following general form

$$\int_0^1 \hat{x}^r(\lambda)\hat{y}^s(\lambda)d\lambda,$$

where $\tilde{x}$ and $\tilde{y}$ are expressed as (4.13a) and (4.13b), respectively. Such 1-D line integrals can be computed either numerically by the Gaussian quadrature rule or even analytically. In [23], the following formulae has been derived to analytically compute the integral in this form:

$$\int_0^1 \hat{x}^r(\lambda)\hat{y}^s(\lambda)d\lambda = \sum_{k=0}^{r+s}\frac{1}{k+1}\Big(\sum_{j=0}^{k} a_{r,j}b_{s,k-j}\Big), \tag{4.15}$$

with

$$a_{r,j} = \begin{pmatrix} r \\ j \end{pmatrix}\hat{x}_1^{r-j}\Delta\hat{x}^j, \qquad b_{s,k-j} = \begin{pmatrix} s \\ k-j \end{pmatrix}\hat{y}_1^{s-(k-j)}\Delta\hat{y}^{k-j},$$

where

$$\Delta\tilde{x}=\hat{x}_2-\hat{x}_1, \qquad \Delta\tilde{y}=\hat{y}_2-\hat{y}_1,$$

and the binomial coefficients are

$$\left(\begin{array}{c} r \\ j \end{array}\right)=\begin{cases} \frac{r!}{j!(r-j)!}, & 0\le j\le r, \\ 0, & \text{otherwise,} \end{cases} \quad \text{and} \quad \left(\begin{array}{c} s \\ k-j \end{array}\right)=\begin{cases} \frac{s!}{(k-j)!(s-(k-j))!}, & 0\le k-j\le s, \\ 0, & \text{otherwise.} \end{cases}$$

Considering the ranges for the values of j in the binomial coefficients, one can rewrite Eq. (4.15) as

$$\int_0^1 \hat{x}^r(\lambda)\hat{y}^s(\lambda)d\lambda=\sum_{k=0}^{r+s}\frac{1}{k+1}\Big(\sum_{j=\max(k-s,0)}^{\min(k,r)} a_{r,j}b_{s,k-j}\Big). \tag{4.16}$$

Eq. (4.16) provides a compact formulae to analytically compute the line integral (4.14) and resolve the difficulties in integrating over general polygons and polygonal cylinders when computing the matrix entries in (2.10a). Furthermore, the analytical formulae eliminates the need of quadrature rules and thus improving the efficiency of the DG-CVS.

5 Boundary condition treatment

When updating solutions at the vertex level, appropriate boundary conditions (b.c.) must be taken into account. Due to the space-time nature of the present DG-CVS, the boundary condition (b.c.) treatment is very simple. For example, the 1-D conservation elements at boundary vertices are shown in Fig. 4. The same convention as in Fig. 3 for denoting the CE boundary sections are used here. Notice that the flux across the outer side boundary (Γ_2 at the left boundary node and Γ_5 at the right boundary node) of the boundary CE is evaluated based on the solution at the new time level. Therefore, the outer side face of the CE has contributions to the matrix M_{ij} (2.10a) instead of the vector b_i (2.10b). In this section, two types of boundary conditions, specified b.c. and outflow b.c., are discussed.

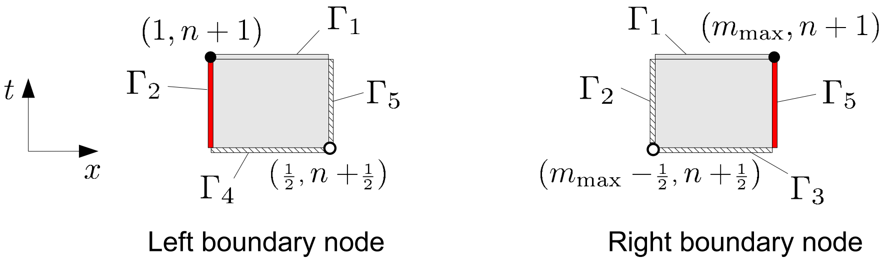


Figure 4: 1-D conservation elements at the boundary node.

5.1 Specified B.C.

Either the solution (Dirichlet type) or its derivative (Neumann type) on the boundary can be specified. In such cases, the matrix shown in (2.10a) can be modified to take into account the specified b.c.. For example, when updating the solution at a space-time boundary node (x_0,t_0), if the solution itself is specified (e.g., inflow b.c.) as u_b, then the linear system can be modified as

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & M_{22} & \cdots & M_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & M_{N2} & \cdots & M_{NN} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_N \end{bmatrix} = \begin{bmatrix} u_b \\ b_2 - M_{21}u_b \\ \vdots \\ b_N - M_{N1}u_b \end{bmatrix}, \tag{5.1}$$

since s_1 in the unknown solution vector represents u_0^h, the solution itself. If the solution's spatial derivative is specified as $(u_x)_b$, then the second row and column of the matrix should be modified, since s_2 represents $(u_x^h)_0$,

$$\begin{bmatrix} M_{11} & 0 & M_{13} & \cdots & M_{1N} \\ 0 & 1 & 0 & \cdots & 0 \\ M_{31} & 0 & M_{33} & \cdots & M_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ M_{N1} & 0 & M_{N3} & \cdots & M_{NN} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_N \end{bmatrix} = \begin{bmatrix} b_1 - M_{12}(u_x)_b \\ (u_x)_b \\ b_3 - M_{32}(u_x)_b \\ \vdots \\ b_N - M_{N2}(u_x)_b \end{bmatrix}. \tag{5.2}$$

5.2 Outflow B.C.

For outflow b.c., nothing needs to be done to modify the matrix as long as the matrix is formed by taking into account the contribution from the outer side face of the boundary CE shown in Fig. 4. The ease of outflow b.c. treatment makes the present space-time DG-CVS very attractive.

6 Numerical tests and discussions

In this section, several numerical tests are presented and the performance of the current DG-CVS will be discussed.

To start the DG-CVS simulation at $t=0$, the solutions at each quadrature point at the bottom face and side faces of the cell-level conservation elements must be known since they are required to evaluate the right hand side vector of the linear system (2.9). Since the initial conditions are often provided as $u(\mathbf{x},0)$ which is a function of spatial coordinates, so there is no difficulty in evaluating u at the quadrature point at the bottom face of the CE. However, at the quadrature points at the side faces of the CE where $t>0$, the solutions are generally unknown, so we simply take the solution at the same spatial location at $t=0$ plane. In the case of Taylor expansions, this is equivalent to assume all temporal derivatives of the solution are zero at $t=0$. This is how we start the simulation in the general case.

However, when we conduct stability study and grid convergence study, we use test cases whose analytical solutions are available. To eliminate the error caused by inaccurate initial conditions, we use the analytical solution to evaluate the solution at the quadrature points at the side faces of the CE.

6.1 Stability study

Here the stability criteria of the present DG-CVS for various degrees of polynomials are determined numerically. The following test case about a linear advection of a sinusoidal wave is used for this purpose

$$\frac{\partial u}{\partial t}+\frac{\partial u}{\partial x}=0, \qquad -1\leq x\leq 1, \tag{6.1a}$$

$$u(x,0)=u_0(x)=\sin(\pi x), \qquad \text{periodic b.c.} \tag{6.1b}$$

The computational domain is spatially discretized by evenly spaced 10 cells. The maximum time step which yields stable solutions (i.e., $||\text{error}||_\infty<1.0$, which is the peak of $u(x,0)$) after 100 periods is recorded for each run. To eliminate the inaccuracy caused by the ill-conditioning of matrix when p is high, all cases are run using quadruple precision via MPACK [18]. The corresponding maximum Courant number

$$\sigma_{\max}=\frac{|a|\delta t_{\max}}{\delta x},$$

where $\delta t_{\max}$ is the maximum time step and δx is the mesh size is indicated in Fig. 5.

Generally, the stable Courant number decreases with increased p. However, Fig. 5 shows that $p3$ and $p4$ have almost the same maximum Courant number. So is true for $p5$ and $p6$. The reason behind this phenomenon remains unclear to us. It might be due to the choice of Taylor polynomials. It is worthy of mention that the original CVS [28] and the CE/SE method [3], which are second order accurate, allows a larger Courant number (i.e., $\sigma\leq 1$) than the $p1$ version of the present DG-CVS (i.e., $\sigma\leq 0.617$). Though the maximum Courant numbers shown in Fig. 5 are obtained from the 1-D linear advection equation, our numerical experiments verify that the same Courant numbers can be used in nonlinear advection equations without causing instability. Therefore, the maximum Courant numbers in Fig. 5 provide a practical guide for choosing a stable time step.

6.2 Sensitivity to the Courant number

It is interesting to test the sensitivity of the present DG-CVS to small Courant numbers. It is well known that the original second-order CE/SE method [4] and the fully discrete N-T scheme [19] exhibit excessive numerical dissipation when small time steps are used.

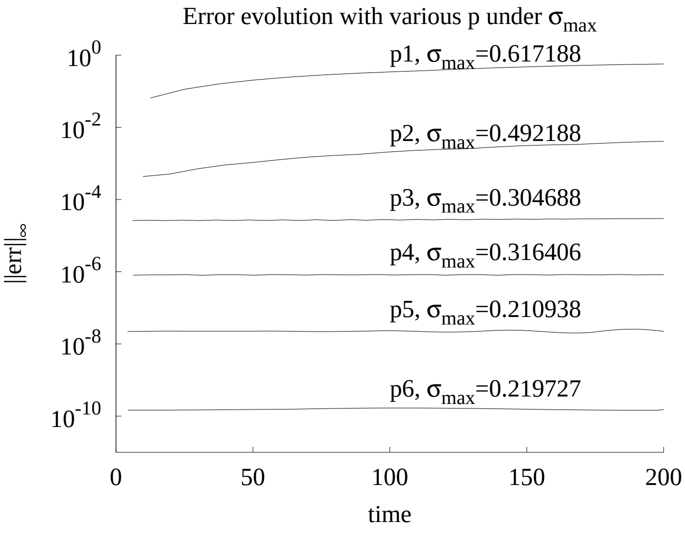


Figure 5: Long time evolution of l_∞ error of the 1-D sinusoidal wave advection under maximum Courant number, $\sigma_{\max}=|a|\delta t_{\max}/\delta x$.

Therefore, special techniques are needed in the CE/SE method and the N-T scheme to mitigate this phenomenon.

In the first test, Eq. (6.1) is solved using DG-CVS with various p and Courant numbers. Table 2 lists the ∞-errors of $p1$, $p2$, $p3$ and $p4$ cases when various Courant numbers are used. It can be seen that the $p1$ case exhibits similar phenomenon to the original CE/SE method, namely, when the time step is vanishingly small, the accuracy is degraded. However, for higher degrees, the DG-CVS is less and less sensitive to small Courant numbers. Actually, in the case of $p4$, a slightly higher accuracy is obtained when a smaller time step is used. This test shows that it is advantageous to employ high-order basis polynomials to avoid the Courant-number-sensitivity problem.

Table 2: Sensitivity to the Courant number on the smooth solution.

$\|\text{error}\|_\infty$ under different Courant number				
Courant number $\sigma=\|a\|\delta t/\delta x$	$p1$	$p2$	$p3$	$p4$
0.2	3.31E-02	1.42E-04	2.62E-05	4.12E-07
0.02	1.82E-01	3.18E-04	2.63E-05	2.42E-07
0.002	8.20E-01	3.14E-03	2.83E-05	2.08E-07

Fig. 6 shows another test on the non-smooth solution. Here, mixed waves with initial solution given by Eq. (6.7) are advected with periodic boundary conditions. Fig. 6 compares the solutions at $t=8.0$ with $\sigma=0.25$ and $\sigma=0.0025$. The comparison clearly verifies the same conclusion from the smooth case: DG-CVS is less and less sensitive to small Courant numbers with increasing p.

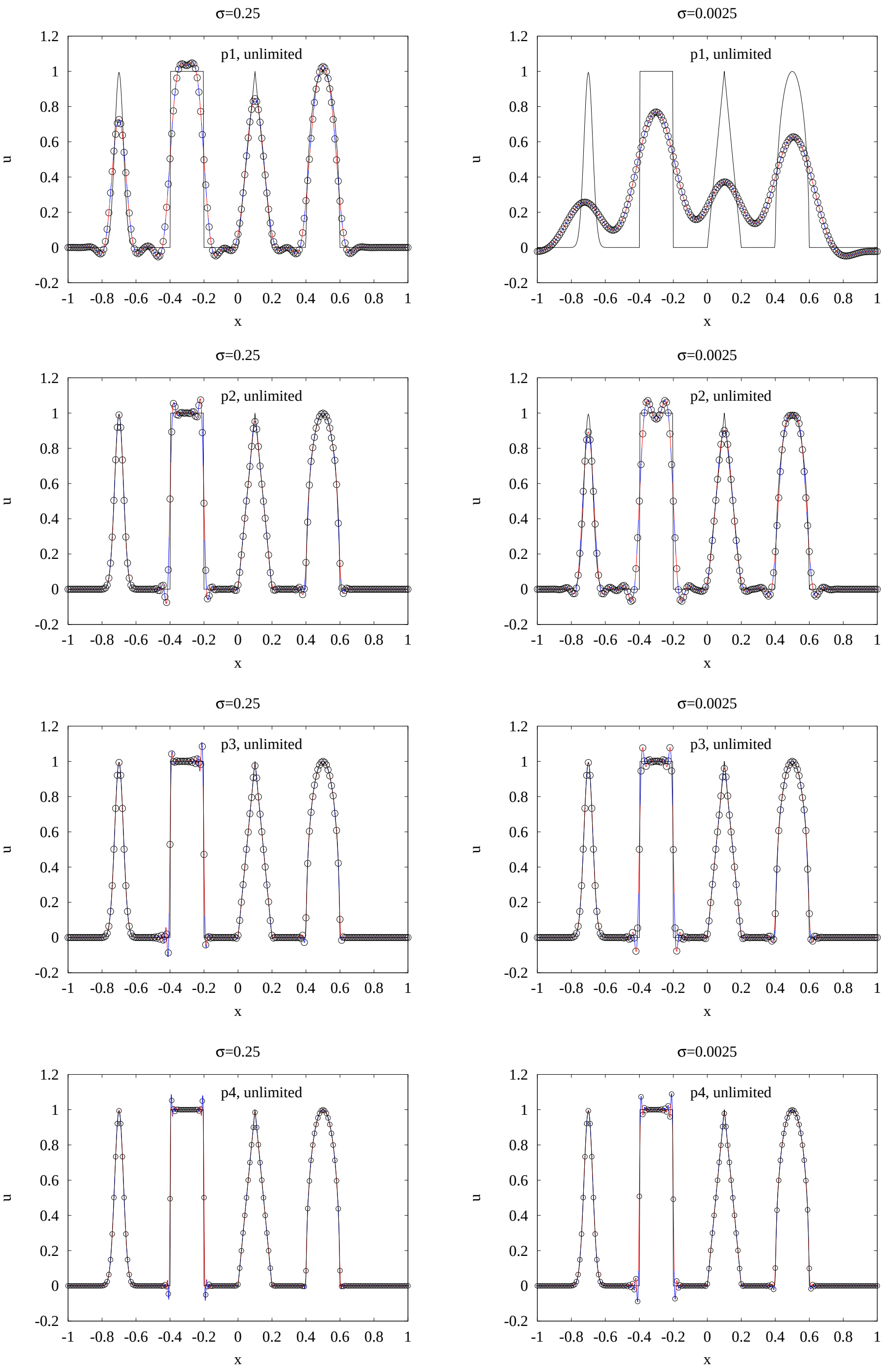


Figure 6: Sensitivity to the Courant number on the non-smooth solution. Solution of the linear advection of mixed waves at $t=8.0$. Left column: $\sigma=0.25$ and right column: $\sigma=0.0025$. Thick solid line: exact solution; circles and thin colorful solid lines: current numerical solutions.

6.3 Grid convergence study

Here the order of accuracy of the present CVS is numerically verified by conducting mesh-refinement study. A sequence of meshes are created by isotropically subdividing the previous coarser spatial mesh. The l_∞ error norm and the l_1 error norm are measured on two successive meshes to derive the convergence rate. The l_∞ error norm and the l_1 error norm are measured as the vertex-based error according to

$$l_\infty(\epsilon) = \max_{i=1}^{n_\text{v}} |u_i^h - u_i^\text{exact}|, \tag{6.2a}$$

$$l_1(\epsilon) = \frac{1}{n_\text{v}} \sum_{i=1}^{n_\text{v}} |u_i^h - u_i^\text{exact}|, \tag{6.2b}$$

where n_v is the number of vertices used to discretize the computational domain. u_i^h is the computed solution and u_i^exact is the exact solution at the ith vertex. The order of accuracy is computed according to

$$\text{order of accuracy} = \frac{\ln\left(\frac{l_k(\epsilon)_{h/2}}{l_k(\epsilon)_h}\right)}{\ln(0.5)}, \tag{6.3}$$

where h denotes the refinement level, namely, $h/2$ is twice as refined as h.

6.3.1 1-D linear advection equation

The first test case is the one described by (6.1). The exact solution for this case is $u(x,t) = \sin(\pi(x-t))$. Tables 3 and 5 summarizes the results by varying the degree (i.e., p) of basis polynomials and the mesh refinement. Here the same Courant number $\sigma = 0.2$ is used in all $p1$-$p6$ cases. All runs reach the final time at $t=1.0$. The $p1$-$p4$ results in Table 3 are obtained using double precision arithmetic. As can be seen, the expected asymptotic orders of accuracy are obtained for all $p1$-$p4$ basis polynomials except the last row ($p=4$, $n_\text{c} = 160$) where the solution error is so small that the condition number of the linear system comes into effect. Table 4 lists the condition number of the linear system for various p. When $p=4$, the condition number is $1.475e5$. For this condition number, the double precision arithmetic is not able to reliably solve the system with error smaller than 10^{-12}. Table 5 shows the $p4$-$p6$ results using quadruple precision arithmetic. As can been seen, correct convergence rate can be obtained even for $p6$ cases. Therefore, the optimal order of accuracy of $p1$, $p2$, $p3$, $p4$, $p5$ and $p6$ cases are 2, 3, 4, 5, 6 and 7, respectively.

6.3.2 1-D non-linear Burgers equation

Here, the nonlinear inviscid Burgers equation is tested,

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right) = 0, \qquad -1 \le x \le 1, \tag{6.4a}$$

$$u(x,0) = -\sin(\pi x), \qquad \text{periodic b.c.} \tag{6.4b}$$

Table 3: Order of accuracy for the 1-D linear sinusoidal wave advection at $t=1.0$. Double precision.

p	n_c	l_1 error	order	l_∞ error	order
1	10	1.96E-02	-	3.31E-02	-
	20	3.93E-03	2.318	6.50E-03	2.347
	40	8.27E-04	2.247	1.33E-03	2.286
	80	1.86E-04	2.155	2.95E-04	2.174
	160	4.37E-05	2.087	6.91E-05	2.097
2	10	9.53E-05	-	1.42E-04	-
	20	1.15E-05	3.050	1.73E-05	3.035
	40	1.42E-06	3.017	2.20E-06	2.980
	80	1.77E-07	3.008	2.75E-07	2.995
	160	2.20E-08	3.005	3.44E-08	2.999
3	10	1.55E-05	-	2.62E-05	-
	20	1.03E-06	3.902	1.71E-06	3.934
	40	6.62E-08	3.965	1.07E-07	4.006
	80	4.19E-09	3.983	6.66E-09	4.002
	160	2.63E-10	3.991	4.16E-10	4.000
4	10	2.80E-07	-	4.12E-07	-
	20	8.18E-09	5.096	1.25E-08	5.037
	40	2.46E-10	5.057	3.81E-10	5.041
	80	7.37E-12	5.060	1.16E-11	5.036
	160	6.40E-13	3.526	1.96E-12	2.564

Table 4: Condition number of the LHS matrix in the 1-D linear advection of the sinusoidal function.

Degree of polynomials (p)	1	2	3	4	5	6
Condition number	6.600E+00	5.450E+01	4.939E+03	1.475E+05	4.410E+06	1.530E+08

Table 5: Order of accuracy for the 1-D linear sinusoidal wave advection at $t=1.0$. Quadruple precision.

p	n_c	l_1 error	order	l_∞ error	order
4	10	2.80E-07	-	4.12E-07	-
	20	8.18E-09	5.096	1.25E-08	5.037
	40	2.46E-10	5.057	3.81E-10	5.041
	80	7.48E-12	5.039	1.17E-11	5.029
	160	2.32E-13	5.013	3.63E-13	5.008
5	10	1.31E-08	-	2.19E-08	-
	20	2.14E-10	5.933	3.53E-10	5.960
	40	3.42E-12	5.967	5.51E-12	6.001
	80	5.39E-14	5.986	8.58E-14	6.004
	160	8.44E-16	5.999	1.33E-15	6.007
6	10	2.00E-11	-	2.95E-11	-
	20	1.43E-13	7.129	2.16E-13	7.094
	40	1.06E-15	7.072	1.64E-15	7.039
	80	8.18E-18	7.022	1.28E-17	7.009
	160	6.38E-20	7.003	9.98E-20	6.997

The solution will remain smooth up to $t=1/\pi$. After that, the solution will evolve into a discontinuity (wave breaking). The analytical solution exists for this case in an implicit form

$$u(x,t)=-\sin(\pi(x-ut)).$$

At time t=0.12, the solution is still smooth and can be used to verify the order of accuracy.

The Newton-Raphson method explained in Subsection 2.5 is used for this nonlinear case. Tables 6 and 7 show similar orders of accuracy to those in Tables 3 and 7. That is, the theoretical asymptotic convergence rates can be obtained for all basis polynomials of various degrees. In Tables 6 and 7, $\bar{n}_{\text{iter}}$ represents the average number of Newton-Raphson iterations at the vertex level to drive the residual (2.11)

$$\|\mathbf{G}(\mathbf{s})\|\le 10^{-14}.$$

It can be seen that 3-6 iterations are typically sufficient for the solution to converge. Again, for high p ($p>4$), quadruple precision arithmetic is needed for accuracy.

6.3.3 2-D linear advection equation

This subsection and next subsection will conduct the grid convergence study on 2-D problems. Fig. 7 shows the coarsest rectangular mesh and unstructured triangular mesh used in the study. The meshes will be refined isotropically several times in the study.

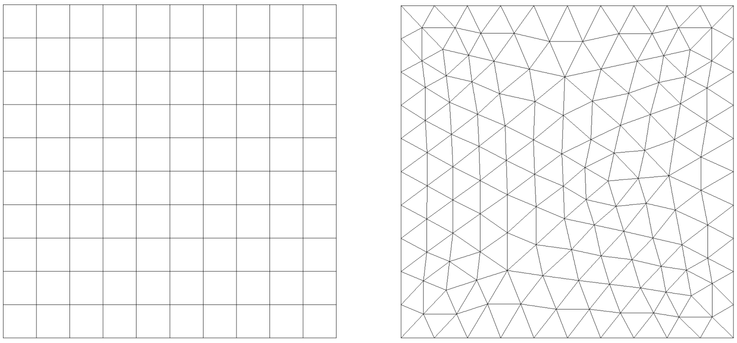


Figure 7: Two-dimensional meshes used in the study of the grid convergence. Left: rectangular mesh. Right: unstructured triangular mesh.

The first 2-D problem is about a sinusoidal surface advected along the diagonal direction of a square region

$$\frac{\partial u}{\partial t}+\frac{\partial u}{\partial x}+\frac{\partial u}{\partial y}=0, \qquad -1\le x\le 1,\ -1\le y\le 1, \tag{6.5a}$$

$$u(x,y,0)=\sin(\pi(x+y)), \qquad \text{periodic b.c.} \tag{6.5b}$$

Due to the ill-conditioning (cf. Table 8) resulted from the high-order Taylor polynomials, we do not attempt to run cases with $p>4$. The results at $t=1.0$ for $p1$-$p4$ cases are

Table 6: Order of accuracy for the 1-D inviscid Burgers equation at $t=0.12$. Double precision.

p	n_c	$\bar{n}_{\text{iter}}$	l_1 error	order	l_∞ error	order
1	10	4.06	1.08E-02	-	3.42E-02	-
	20	3.97	3.07E-03	1.814	8.25E-03	2.054
	40	3.85	7.78E-04	1.980	2.04E-03	2.013
	80	3.65	1.93E-04	2.009	4.87E-04	2.069
	160	3.09	4.81E-05	2.009	1.19E-04	2.039
2	10	4.18	6.84E-05	-	1.45E-04	-
	20	3.90	1.13E-05	2.597	2.96E-05	2.287
	40	3.88	1.36E-06	3.060	3.49E-06	3.086
	80	3.57	1.71E-07	2.984	4.05E-07	3.108
	160	3.01	2.12E-08	3.011	4.87E-08	3.055
3	10	4.48	5.33E-05	-	2.13E-04	-
	20	4.00	5.25E-06	3.344	3.17E-05	2.753
	40	3.94	3.35E-07	3.972	1.97E-06	4.003
	80	3.79	2.08E-08	4.008	1.23E-07	4.005
	160	3.42	1.29E-09	4.012	7.83E-09	3.973
4	10	4.52	5.57E-06	-	2.16E-05	-
	20	4.02	1.79E-07	4.958	8.31E-07	4.703
	40	3.93	8.09E-09	4.469	4.30E-08	4.270
	80	3.78	2.47E-10	5.032	1.60E-09	4.750
	160	3.39	7.69E-12	5.006	5.32E-11	4.911

Table 7: Order of accuracy for the 1-D inviscid Burgers equation at $t=0.12$. Quadruple precision.

p	n_c	$\bar{n}_{\text{iter}}$	l_1 error	order	l_∞ error	order
4	10	5.45	5.57E-06	-	2.16E-05	-
	20	5.00	1.79E-07	4.958	8.31E-07	4.703
	40	4.93	8.09E-09	4.469	4.30E-08	4.270
	80	4.82	2.47E-10	5.032	1.60E-09	4.750
	160	4.56	7.60E-12	5.023	5.31E-11	4.912
5	10	5.64	1.44E-06	-	6.93E-06	-
	20	5.21	1.84E-08	6.296	1.16E-07	5.905
	40	4.98	5.33E-10	5.108	6.81E-09	4.086
	80	4.89	9.86E-12	5.756	1.18E-10	5.848
	160	4.74	1.67E-13	5.882	1.89E-12	5.968
6	10	5.61	4.98E-07	-	1.80E-06	-
	20	5.30	6.82E-09	6.188	3.23E-08	5.799
	40	4.96	5.87E-11	6.862	3.55E-10	6.510
	80	4.90	5.15E-13	6.832	4.46E-12	6.312
	160	4.77	4.61E-15	6.803	4.63E-14	6.592

Table 8: Condition number of the LHS matrix in the 2-D linear advection of the sinusoidal function.

Degree of polynomials (p)	1	2	3	4
Condition number	6.000E+00	1.360E+02	3.856E+03	1.355E+05

Table 9: Order of accuracy for the 2-D linear sinusoidal wave advection at $t=1.0$ on rectangular meshes.

p	mesh size	l_1 error	order	l_∞ error	order
1	10×10	5.77E-02	-	8.98E-02	-
	20×20	1.04E-02	2.47	1.65E-02	2.44
	40×40	1.98E-03	2.39	3.11E-03	2.41
	80×80	4.12E-04	2.26	6.47E-04	2.27
	160×160	9.24E-05	2.16	1.45E-04	2.16
2	10×10	3.59E-04	-	5.58E-04	-
	20×20	2.41E-05	3.9	3.79E-05	3.88
	40×40	3.24E-06	2.89	5.08E-06	2.90
	80×80	4.16E-07	2.96	6.53E-07	2.96
	160×160	5.23E-08	2.99	8.21E-08	2.99
3	10×10	1.92E-04	-	2.99E-04	-
	20×20	1.17E-05	4.04	1.85E-05	4.01
	40×40	7.09E-07	4.04	1.11E-06	4.06
	80×80	4.38E-08	4.02	6.88E-08	4.01
4	10×10	1.54E-05	-	2.38E-05	-
	20×20	5.29E-07	4.86	8.33E-07	4.84
	40×40	1.70E-08	4.96	2.67E-08	4.96
	80×80	5.35E-10	4.99	8.40E-10	4.99

Table 10: Order of accuracy for the 2-D linear sinusoidal wave advection at $t=1.0$ on unstructured triangular meshes.

p	mesh size (h)	l_1 error	order	l_∞ error	order
1	0.2	5.21E-02	-	1.07E-01	-
	0.1	9.14E-03	2.51	2.01E-02	2.41
	0.05	1.70E-03	2.43	3.97E-03	2.34
	0.025	3.48E-04	2.29	8.37E-04	2.25
	0.0125	7.71E-05	2.17	1.91E-04	2.13
2	0.2	4.68E-04	-	1.90E-03	-
	0.1	4.06E-05	3.53	2.94E-04	2.69
	0.05	3.88E-06	3.39	3.60E-05	3.03
	0.025	4.09E-07	3.25	4.46E-06	3.01
	0.0125	4.60E-08	3.15	5.60E-07	2.99
3	0.2	7.05E-05	-	2.77E-04	-
	0.1	4.48E-06	3.98	1.76E-05	3.98
	0.05	2.84E-07	3.98	1.11E-06	3.99
	0.025	1.80E-08	3.98	6.96E-08	4.00
4	0.2	1.91E-06	-	8.61E-06	-
	0.1	5.79E-08	5.04	2.80E-07	4.94
	0.05	1.75E-09	5.05	9.41E-09	4.90
	0.025	5.39E-11	5.02	3.18E-10	4.89

tabulated in Tables 9 and 10 for rectangular meshes and triangular meshes, respectively. The time steps are $\delta t=0.05$ and $\delta t=0.025$ for the coarsest rectangular mesh and triangular mesh, respectively. In the case of rectangular meshes, this time step corresponds to Courant number $\sigma = 0.25$ for all runs. Again, the theoretical convergence rates have been obtained for all runs. The quadrature-free approach explained in Section 4 was used to compute the matrix entries in (2.10a). This example demonstrates that the current method is also suitable for unstructured meshes with high accuracies.

6.3.4 2-D non-linear inviscid Burgers equation

The second 2-D case is the two dimensional inviscid Burgers equation defined as

$$\frac{\partial u}{\partial t}+\frac{\partial}{\partial x}\Big(\frac{1}{2}u^2\Big)+\frac{\partial}{\partial y}\Big(\frac{1}{2}u^2\Big)=0, \qquad -1\le x\le 1,\ -1\le y\le 1, \tag{6.6a}$$

$$u(x,y,0)=-\sin(\pi(x+y)), \qquad \text{periodic b.c.} \tag{6.6b}$$

The same accuracy study as in the 1-D Burgers equation is conducted for this case. The time steps are $\delta t = 0.04$ and $\delta t = 0.025$ for the coarsest rectangular mesh and triangular mesh, respectively. Tables 11 and 12 shows similar orders of accuracy to the 1D cases. Due to the non-linear nature of the problem, the convergence rates are not as neat as the linear case, but nevertheless show the correct trend toward optimal rates. As can also be seen that 2-5 iterations are sufficient to drive the residual below 10^{-14} in each time step.

Table 11: Order of accuracy for the 2-D inviscid Burgers equation at $t = 0.08$ on rectangular meshes.

p	mesh size	$\bar{n}_{\text{iter}}$	l_1 error	order	l_∞ error	order
1	10×10	3.55	2.49E-02	-	8.97E-02	-
	20×20	3.58	7.45E-03	1.74	3.14E-02	1.51
	40×40	3.56	1.89E-03	1.98	7.17E-03	2.13
	80×80	3.10	4.60E-04	2.04	1.62E-03	2.15
	160×160	2.96	1.12E-04	2.04	3.84E-04	2.08
2	10×10	3.51	8.27E-04	-	2.67E-03	-
	20×20	3.58	8.88E-05	3.22	4.98E-04	2.42
	40×40	3.50	5.12E-06	4.12	1.67E-05	4.9
	80×80	3.01	8.60E-07	2.57	4.88E-06	1.77
	160×160	2.96	9.59E-08	3.16	5.23E-07	3.22
	320×320	2.96	1.07E-08	3.16	4.91E-08	3.41
3	10×10	3.80	3.26E-04	-	8.02E-04	-
	20×20	3.65	1.07E-04	1.61	7.89E-04	0.02
	40×40	3.64	9.04E-06	3.57	7.76E-05	3.35
	80×80	3.57	5.99E-07	3.92	5.17E-06	3.91
	160×160	2.96	3.73E-08	4.01	3.37E-07	3.94
4	10×10	3.84	2.54E-04	-	9.31E-04	-
	20×20	3.70	5.90E-06	5.43	2.64E-05	5.14
	40×40	3.64	3.87E-07	3.93	4.37E-06	2.59
	80×80	3.17	1.31E-08	4.88	1.23E-07	5.15
	160×160	2.96	4.59E-10	4.83	3.96E-09	4.96

Table 12: Order of accuracy for the 2-D inviscid Burgers equation at $t=0.075$ on unstructured triangular meshes.

p	mesh size (h)	$\bar{n}_{\text{iter}}$	l_1 error	order	l_∞ error	order
1	0.2	3.39	2.12E-02	-	9.92E-02	-
	0.1	3.53	5.58E-03	1.93	2.79E-02	1.83
	0.05	3.18	1.40E-03	1.99	7.26E-03	1.94
	0.025	2.95	3.46E-04	2.02	1.75E-03	2.05
	0.0125	2.96	8.59E-05	2.01	4.26E-04	2.04
2	0.2	3.38	7.50E-04	-	1.14E-02	-
	0.1	3.55	1.03E-04	2.86	1.73E-03	2.72
	0.05	3.17	1.19E-05	3.11	2.60E-04	2.73
	0.025	2.94	1.21E-06	3.30	4.21E-05	2.63
	0.0125	2.97	1.28E-07	3.24	7.14E-06	2.56
3	0.2	3.52	4.35E-04	-	6.34E-03	-
	0.1	3.64	3.47E-05	3.65	3.85E-04	4.04
	0.05	3.29	2.42E-06	3.84	4.00E-05	3.27
	0.025	2.95	1.52E-07	3.99	2.73E-06	3.87
	0.0125	2.97	9.55E-09	3.99	1.67E-07	4.03
4	0.2	3.58	6.10E-05	-	7.15E-04	-
	0.1	3.62	2.71E-06	4.49	3.82E-05	4.23
	0.05	3.26	8.18E-08	5.05	1.77E-06	4.43
	0.025	2.94	2.36E-09	5.12	8.07E-08	4.46

6.4 Discontinuity capturing

To demonstrate the discontinuity capturing capability of the DG-CVS, the problem about the advection of mixed Gaussian, square, triangle and elliptical waves [11] is tested. The initial condition of the problem is defined as follows:

$$u(x,0)=\begin{cases} \frac{1}{6}\big(G(x,\beta,b-\delta)+G(x,\beta,b+\delta)+4G(x,\beta,b)\big), & -0.8\le x\le -0.6,\\ 1, & -0.4\le x\le -0.2,\\ 1-|10(x-0.1)|, & 0\le x\le 0.2,\\ \frac{1}{6}\big(E(x,\alpha,a-\delta)+E(x,\alpha,a+\delta)+4E(x,\alpha,a)\big), & 0.4\le x\le 0.6,\\ 0, & \text{otherwise,}\end{cases} \tag{6.7}$$

where $a=0.5$, $b=-0.7$, $\delta=0.005$, $\alpha=10$ and $\beta=\log 2/(36\delta^2)$. G and E are Gaussian and elliptical function, respectively, where

$$\begin{aligned} G(x,\beta,b)&=e^{-\beta(x-b)^2},\\ E(x,\alpha,a)&=\sqrt{\max\{1-\alpha^2(x-a)^2,0\}}. \end{aligned}$$

The spatial computational domain is $[-1,1]$. Again, periodic boundary conditions are assumed at the ends of the domain.

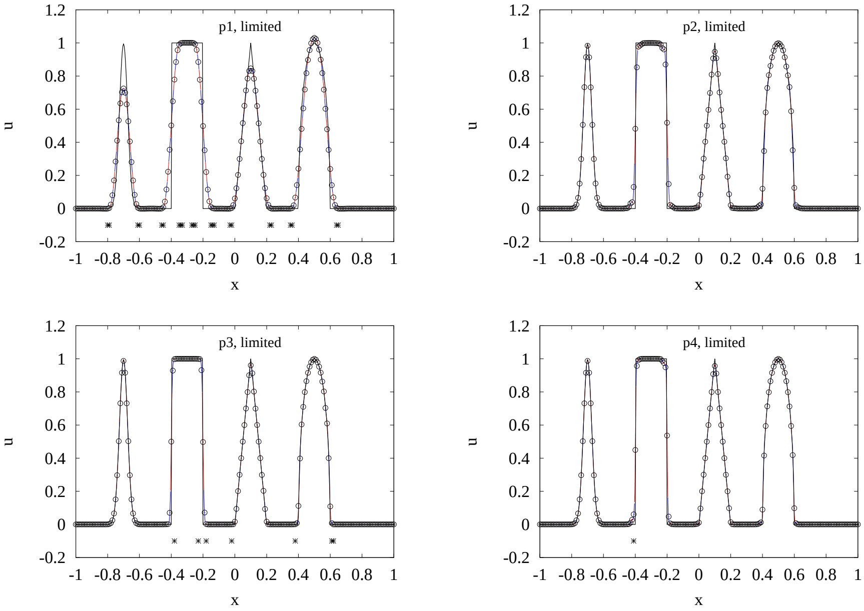


Figure 8: Linear advection of mixed waves. Limited solution at $t=8.0$. The computational domain consists of 200 uniform elements. Thick solid line: exact solution; circles and thin colorful solid lines: current numerical solutions; and stars: oscillation indicators.

The computational domain is discretized by 200 evenly spaced cells which is a pretty coarse mesh for this case. The time step used in all simulations is $\delta t = 0.0025$ corresponding to the Courant number $\sigma = 0.25$. Fig. 8 shows the limited solutions at $t = 8.0$ using basis polynomials of various degrees. As can be seen, both the discontinuities and smooth regions are better captured with increasing p. With increasing p, the transition of discontinuities is steeper. The second order run ($p1$) is not able to resolve the Gaussian wave which is steep and smooth.

The limiter adopted here is described in the author's another article [27]. In the limiting procedure, the oscillation regions are first detected using an indicator. Then the limiter is applied only in the indicated region and suppress the oscillations. As can be seen, compared to the corresponding unlimited solutions, the oscillations around the discontinuities have been successfully suppressed for all cases. Comparing with the non-limited solution (cf. Fig. 6 left column), one can see that the peaks of smooth waves are not modified indicating the limiter is not activated at those regions.

6.5 Boundary conditions

Here a doubly raised cosine function is advected according to

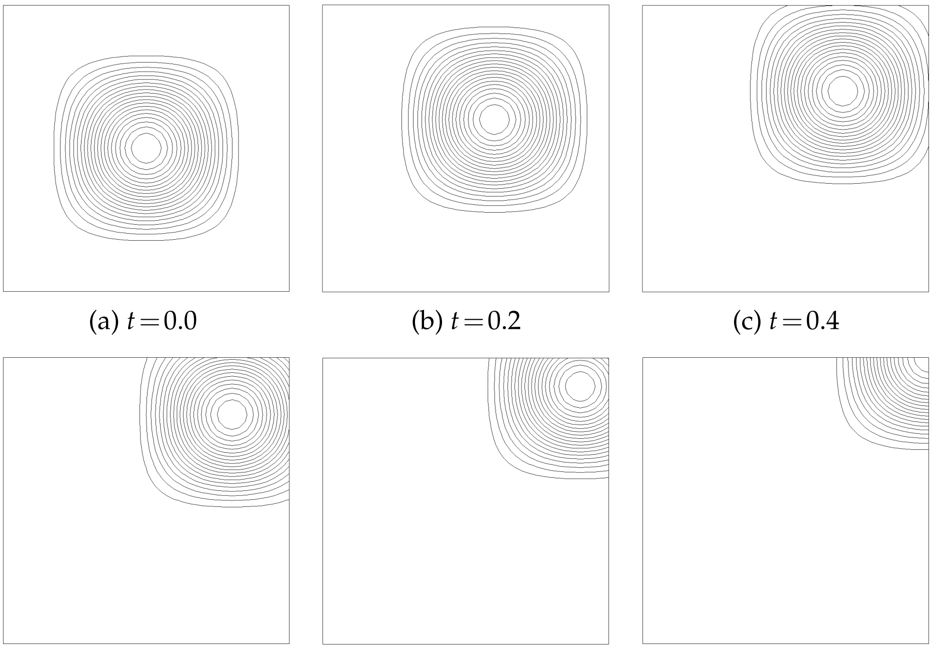
$$\frac{\partial u}{\partial t}+\frac{\partial u}{\partial x}+\frac{\partial u}{\partial y}=0, \qquad -1\le x\le 1, \ -1\le y\le 1, \tag{6.8a}$$

$$u(x,y,0)=\frac{1}{4}\Big[1+\cos\Big(\frac{4\pi}{3}x\Big)\Big]\Big[1+\cos\Big(\frac{4\pi}{3}y\Big)\Big],\quad u(-1,y,t)=u(x,-1,t)=0, \tag{6.8b}$$

and

$$\text{outflow at } x=1 \text{ and } y=1. \tag{6.8c}$$

This problem contains both the specified inflow b.c. and outflow b.c.. The problem is simulated up to the time $t=1.0$ using $p2$ basis polynomials on a 40×40 rectangular mesh. The Courant number used is $\sigma=0.25$. Fig. 9 shows the advected solution at several instants. As can be seen from the contour lines, the cosine surface exits the outflow boundaries smoothly. This example verifies the efficacy of the approach, explained in Section 5, to handle various boundary conditions.



(a) $t=0.0$ (b) $t=0.2$ (c) $t=0.4$

(d) $t=0.6$ (e) $t=0.8$ (f) $t=1.0$

Figure 9: Advection of a doubly raised cosine surface. $p2$ solution on 40×40 rectangular mesh.

6.6 Computational cost comparison

For rectangular meshes, one can choose the Gaussian quadrature rule or the quadrature-free method (see Section 4) to compute the matrix elements (see Eq. (2.10a)) at both the cell level and the vertex level. Both the top face and the volume itself of the CE contribute to the matrix.

To demonstrate the cost saving by using the quadrature-free approach, we run the 2-D advection problem given by (6.5) on a 40×40 rectangular mesh for 80 time steps using basis polynomials of various degrees (p) and record the time cost. In Table 13, we compare the relative costs of the two methods. The number of quadrature points is equal to $(p+1)^2$ and $(p+1)^3$ for the surface integral and the volume integral, respectively. The costs are normalized such that the cost using the quadrature-free approach is unity. As can be seen, the cost saving by using the quadrature-free method is more and more significant with increasing p. For example, when computing the volume integrals, the Gaussian quadrature rule is about 60 times more costly than the quadrature-free approach in the $p4$ case. This simple test clearly shows that the quadrature-free method explained in Section 4 not only makes the integration over general polygons and polygonal cylinders possible, but also save the computational cost significantly.

Table 13: Comparison of the time cost in integration using the Gaussian quadrature rule and quadrature-free approach.

		Scaled time cost			
		$p1$	$p2$	$p3$	$p4$
Compute M_{ij} contributions from the top face of CEs	*Gaussian quadrature*	0.99	1.26	2.08	4.34
	Quadrature-free	1	1	1	1
Compute M_{ij} contributions from the volume of CEs	*Gaussian quadrature*	1.78	5.64	24.48	60.38
	Quadrature-free	1	1	1	1

It is also interesting to compare the relative time costs when basis polynomials of various degrees are used. Table 14 shows the comparison results. In the table, the costs have been normalized such that the cost of the $p1$ case is unity. As can be seen, for both the rectangular mesh and the triangular mesh, $p2$, $p3$ and $p4$ are about 1.5, 3 and 6 times as expensive as $p1$, respectively. Note that the number of unknowns at each space-time node is 4, 10, 20 and 35 for $p1$, $p2$, $p3$ and $p4$ cases, respectively.

Table 14: Comparison of the time cost per timestep per vertex.

	Scaled time cost			
	$p1$	$p2$	$p3$	$p4$
cost per timestep per vertex (triangular mesh)	1	1.54	2.93	6.39
cost per timestep per vertex (rectangular mesh)	1	1.41	2.80	6.40

Finally, we investigate the time cost of individual components in the solution process. Table 15 shows the result in percentages. From the table, it can be seen that the time cost spent in the computation of the right hand side vector b_i (see Eq. (2.10b)) is about 50% of the total cost for all p's. Note that we are using the Gaussian quadrature rule to compute the integrals contributing to b_i. This shows that it is necessary to adopt the quadrature-free method to greatly reduce the cost. The percentage of the cost to compute M_{ij} (see Eq. (2.10a)) decreases with increasing p thanks to the use of the quadrature-free method.

Table 15: Comparison of time cost in forming the matrix and the right-hand-side vector.

	Scaled time cost (%)			
	$p1$	$p2$	$p3$	$p4$
Triangular mesh				
Computation of b_i	50.77	58.06	58.55	51.28
Computation of M_{ij}	20.63	15.44	11.57	9.68
Linear equation solver	7.35	11.86	21.34	33.94
other cost	21.25	14.63	8.55	5.10
Total	100	100	100	100
Rectangular mesh				
Computation of b_i	40.83	46.37	49.08	43.49
Computation of M_{ij}	24.25	19.60	14.00	11.23
Linear equation solver	8.61	14.76	26.06	39.36
other cost	26.32	19.27	10.85	5.92
Total	100	100	100	100

The percentage of the cost of linear equation solver increases with p. The "other cost" represents all other costs except those listed in the table, which decreases with p.

6.7 Solving compressible Euler equations

Though the DG-CVS formulation described in this paper is for scalar linear or nonlinear advection problems. However, the same idea equally applies to more complex hyperbolic nonlinear equation systems, for instance, compressible Euler equations. In our implementation, following Lowrie et al. [17], we choose the working variables to be

$$\mathbf{Q}=\left[\sqrt{\rho},\sqrt{\rho}u,\sqrt{\rho}v,\sqrt{\rho}H\right]^{\mathrm{T}}, \tag{6.9}$$

where ρ,u,v,H are density, x- and y-velocity components, and specific total enthalpy, respectively. By choosing such working variables, all components of the conservative state vector and flux vectors can be expressed as the *double product* between working variables, which is analogous to the inviscid Burger's equation. Here we include some results for the compressible Euler equations which have been presented in our recent AIAA conference paper [26].

6.7.1 1-D Sod's shock-tube problem

This well-known Sod shock tube problem has been extensively used to verify a numerical scheme for unsteady inviscid flows. The initial conditions are a high-pressure region on the left and a low-pressure region on the right separated by a diaphragm located at $x=0.5$. At time $t=0$, the diaphragm is broken, a shock wave will propagate to the right and an expansion wave propagate to the left. There is a contact surface in between across which the density and entropy are discontinuous while the pressure and velocity are continuous. The spatial domain $[0,1]$ is discretized into 81 equally spaced elements.

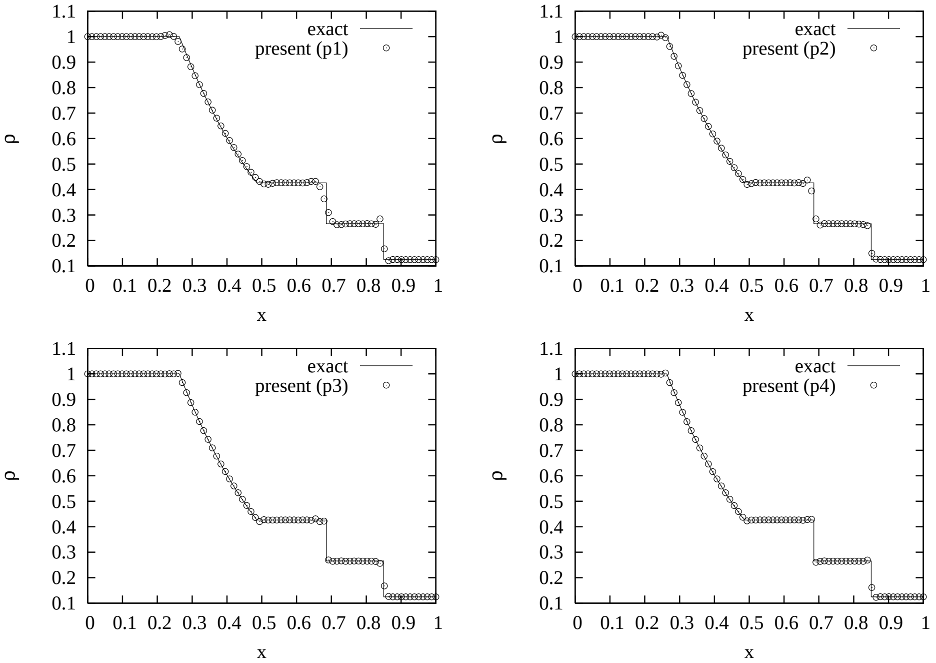


Figure 10: Density distribution of the 1-D shock tube problem at t=0.2 when various degrees of basis polynomials employed. Without limiter.

Fig. 10 shows the density and pressure distribution at $t=0.2$ when basis polynomials of various degrees ($p1$-$p4$) are employed. Note that for all simulations, no any form of limiters are employed to suppress the unphysical overshoots/undershoots. As can be seen, the present space-time Riemann-solver-free method does not produce the expansion shock phenomenon. This indicates that the present scheme automatically satisfies the entropy condition. With increasing degrees of the basis polynomials, the captured shock discontinuity and contact discontinuity are sharper. At higher p, the contact discontinuity can be captured as sharply as the shock wave.

6.7.2 Advection of 2-D isentropic vortex

This example is an isentropic vortex advection problem [25] on a 2D computation domain $[0,10]\times[0,10]$. The initial conditions are given as an isentropic vortex with the center at (5,5), i.e.,

$$u(x,y,0)=1-\frac{\epsilon}{2\pi}e^{0.5(1-r^2)}(y-5), \tag{6.10a}$$

$$v(x,y,0)=1+\frac{\epsilon}{2\pi}e^{0.5(1-r^2)}(x-5), \tag{6.10b}$$

$$T(x,y,0)=1-\frac{(\gamma-1)\epsilon^2}{8\gamma\pi^2}e^{1-r^2}, \quad S(x,y,0)=1, \tag{6.10c}$$

where $u,v,T,$and S are x-velocity, y-velocity, temperature and entropy, respectively. $\epsilon=5$ represents the vortex strength and $r^2=(x-5)^2+(y-5)^2$. The periodic boundary conditions on both directions are assumed. The density ρ and the pressure p can be obtained via

$$\rho(x,y,0)=\Big(\frac{T(x,y,0)}{S(x,y,0)}\Big)^{\frac{1}{\gamma-1}}, \qquad p(x,y,0)=\rho(x,y,0)T(x,y,0),$$

where γ is the ratio of specific heats.

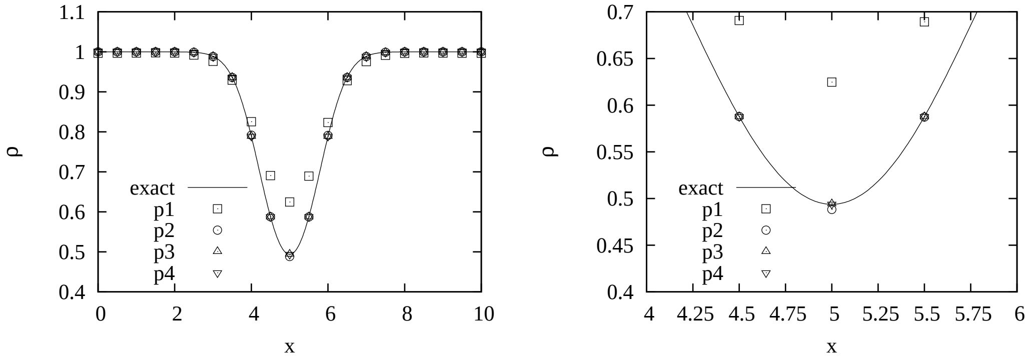


Figure 11: Advection of isentropic vortex on a 20×20 rectangular mesh. Comparison of density accuracy between $p1$-$p4$ cases. The close-up view is on the right.

It can be verified that the Euler equations with the above initial and boundary conditions allows an exact solution which is the initial solution advected with the speed (1,1) in the diagonal direction. Fig. 11 shows the comparison between $p1$, $p2$, $p3$, and $p4$ density distributions at the horizontal cut $y=5$ at the end of one period. The underlying mesh is a 20×20 rectangular mesh. The exact solution is also shown in the same figure for comparison. As can be seen, on this coarse mesh, $p1$ simulation does not yield satisfactory results compared with the exact solution. Higher accuracy is obtained with increasing p. The maximum errors shown in Table 16 further confirm the superior accuracy when higher degree of basis polynomials are used.

Table 16: Maximum density error in the case of advection of isentropic vortex on a 20×20 rectangular mesh.

	$p1$	$p2$	$p3$	$p4$
$\lvert u^{\text{exact}}-u^h\rvert$ at location (5,5)	1.31E-01	7.50E-03	5.26E-04	9.30E-06

6.7.3 Supersonic forward-facing problem

This is another benchmark problem to verify a time accurate inviscid compressible flow solver. The problem is a supersonic flow with $M=3$ flows through a channel of length 3 and height 1. A step of height 0.2 stands at location 0.6 downstream of the inlet. The step acts as an obstacle to the supersonic inflow. A transient detached shock wave will

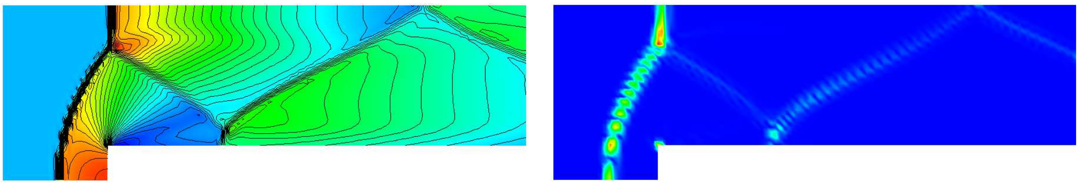


Figure 12: Solution of a supersonic ($M=3$) flow through a channel with forward-facing step at $t=4.0$. Without limiter. Left: $p1$ density field. Right: residual of the continuity equation.

be formed and hit the top wall. The shock wave will reflect from the top wall and further hit the bottom wall and reflect again. The flow expands rapidly at the singularity corner.

In the present simulation, a rectangular mesh of $\delta x=\delta y=1/40$ is used. The linear basis polynomials ($p1$) is employed (higher p requires solution limiting for robustness, which has not been implemented yet). Fig. 12 (left) shows the density field at $t=4.0$. As can be seen, the main features including the detached bow shock, the Mach stem below the top wall, slip line, reflected shock waves are well captured. Here, no any type of boundary conditions is applied at the singularity corner and no numerical difficulty was encountered, at least for this $p1$ case. No expansion shocks are seen around the corner. In addition, the present method does not exhibit the so-called car-buncle problem in the subsonic region behind the bow shock. It is also interesting to show the residual of the continuity equation in Fig. 12 (right). Since in DG-CVS, the space-time conservation is enforced in the weak form of the governing equations, the resulting solution may not exactly satisfy the original differential governing equations, especially in non-smooth regions as shown here. The residual can be used as an effective smoothness indicator. One may utilize this indicator to implement hp-adaptation to improve the accuracy and efficiency of the solver.

7 Conclusions

In this paper, a high-order space-time method for hyperbolic conservation laws is described in detail. The method is high order accurate in both space and time. It is composed of two important ingredients. The first is the idea of enforcing the space-time flux conservation on the staggered space-time mesh in the Conservation Element/Solution Element (CE/SE) method. The second is the local discontinuous basis functions of the space-time discontinuous Galerkin (DG) finite element method. The former results in a Riemann-solver-free approach for hyperbolic problems. The latter provides a way to achieve high-order accuracy on a compact stencil. The new scheme offers an approach for the CE/SE method to increase its order of accuracy. In addition, the new scheme provides a Riemann-solver-free alternative for the space-time DG method. The new method updates the solutions at the cell level and the vertex level alternately within each time step. For this cell-vertex solution updating strategy and the method's DG ingredient, the new method is termed as discontinuous Galerkin cell vertex scheme (DG-CVS).

In addition to the detailed formulation, some implementation issues including the choice of basis polynomials, quadrature-free integration and boundary condition treatment are also discussed in this paper.

The most distinct features of the DG-CVS can be summarized as:

- locally and globally space-time flux conservative;
- alternate solution updating at the cell level and the vertex level within each physical time step;
- Riemann-solver-free for advection problem;
- high-order accurate in both space and time;
- highly compact regardless of the order;
- suitable for arbitrarily unstructured meshes;
- simple boundary condition treatment.

Based on the numerical tests provided in this paper, the DG-CVS can be used to effectively solve scalar advection equations and the compressible Euler equations with high accuracy. It is expected that the same method can be used to solve other conservation laws such as the shallow water equations and Maxwell's equations. Due to its space-time and Riemann-solver-free nature, the DG-CVS is especially attractive for the following problems: (i) moving boundary problems and (ii) conservation laws where the eigen structure is not explicitly known.

To make the DG-CVS a competitive method for conservation laws, the following issues should be addressed in future work:

• *accuracy*. The present implementation of DG-CVS uses Taylor polynomials as the basis functions. It has been shown that high-order Taylor polynomials generate severely ill-conditioned system which cannot be solved reliably using double precision arithmetic. Future work should attempt other basis polynomials that will yield well-conditioned system for better accuracy.

• *efficiency*. The present space-time method introduces a lot more unknowns than semi-discrete methods and solves a local system whose size is the number of local unknowns for each mesh node. Therefore, the present method is costly especially when solving the nonlinear problems since iterations are needed. When extended to 3-D problems, the method becomes even more expensive. Thus improving the efficiency is critical in making the DG-CVS a practical method. The quadrature-free integration described in this paper is especially suitable for integrals involving Taylor polynomials. What if other types of polynomials are used as the basis function? This remains to be answered in the future study. Though the DG-CVS is designed for arbitrarily unstructured meshes, using quadrilateral or rectangular meshes yields superior efficiency and accuracy. Our preliminary implementation on overset Cartesian/quadrilateral meshes [29] needs a more accurate solution transfer strategy between meshes to ensure the global flux conservation on overset meshes.

- *robustness*. It is well known that high order methods lack robustness across high gradient regions. Some type of slope or flux limiter is typically used to suppress the unphysical overshoots/undershoots to ensure the robustness. However, high-order solution limiting is far from mature and many researchers are working on this topic.

Acknowledgments

This work is supported by the U.S. Air Force Office of Scientific Research (AFOSR) Computational Mathematics Program under the Award No. FA9550-08-1-0122. The authors are grateful to the School of Engineering and the Department of Computer Engineering at Jackson State University for their support. The authors also wish to thank the anonymous referees for their constructive and insightful comments and suggestions.

References

[1] T. Arbogast and M. Wheeler, A characteristics-mixed finite element method for advection-dominated transport problems, SIAM J. Numer. Anal., 32 (1995), 404–424.

[2] S.-C. Chang, Courant number and Mach number insensitive CE/SE Euler solvers, AIAA Paper 2005–4355,

[3] S.-C. Chang and W. To, A new numerical framework for solving conservation laws: the method of space-time conservation element and solution element, NASA. TM., 1991–104495.

[4] S.-C. Chang and X. Wang, Multi-dimensional Courant number insensitive CE/SE Euler solvers for applications involving highly nonuniform meshes, AIAA Paper 2003–5285.

[5] S.-C. Chang, X. Wang and C. Chow, The space-time conservation element and solution element method: a new high-resolution and genuinely multidimensional paradigm for solving conservation laws, J. Comput. Phys., 156(1) (1999), 89–136.

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

[7] B. Cockburn and C.-W. Shu, Runge-Kutta discontinuous Galerkin methods for convection-dominated problems, J. Sci. Comput., 16(3) (2001), 173–261.

[8] G. Dasgupta, Integration within polygonal finite elements, J. Aerosp. Engrg., 16(1) (2003), 9–18.

[9] C. Dawson, Godunov-mixed methods for advection-diffusion equations in multidimensions, SIAM J. Numer. Anal., 30 (1993), 1315–1332.

[10] J. Gressier and J. Moschetta, Robustness versus accuracy in shock-wave computations, Int. J. Numer. Meth. Fluids., 33(1) (2000), 313–332.

[11] G. Jiang and C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys., 126 (1996), 202–228.

[12] A. Kurganov and E. Tadmor, New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations, J. Comput. Phys., 160(1) (2000), 241–282.

[13] M. Lesoinne and C. Farhat, Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations, Comput. Meth. Appl. Mech. Eng., 134(1) (1996), 71–90.

[14] X.-D. Liu, S. Osher and T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys., 115(1) (1994), 200–212.

[15] Y. Liu, Central schemes on overlapping cells, J. Comput. Phys., 209(1) (2005), 82–104.

[16] C. Y. Loh, L. S. Hultgren and S.-C. Chang, Wave computation in compressible flow using space-time conservation element and solution element method, AIAA J., 39(5) (2001), 794–801.

[17] R. Lowrie, P. Roe, and B. van Leer, A space-time discontinuous Galerkin method for the time-accurate numerical solution of hyperbolic conservation laws, AIAA Paper 1995–1658.

[18] M. Nakata, The MPACK: Multiple Precision Arithmetic BLAS (MBLAS) and LAPACK (MLAPACK), 2009, `http://mplapack.sourceforge.net/`.

[19] H. Nessyahu and E. Tadmor, Non-oscillatory central differencing for hyperbolic conservation laws, J. Comput. Phys., 87 (1990), 408–463.

[20] J. Palaniappan, R. B. Haber and R. L. Jerrard, A spacetime discontinuous Galerkin method for scalar conservation laws, Comput. Meth. Appl. Mech. Eng., 193(33-35) (2004), 3607–3631.

[21] M. Pandolfi and D. D'Ambrosio, Numerical instablilities in upwind methods: analysis and cures for the "carbuncle" phenomenon, J. Comput. Phys., 166(2) (2001), 271–301.

[22] A. Patera, A spectral element method for fluid dynamics-laminar flow in a channel expansion, J. Comput. Phys., 54 (1984), 468–488.

[23] H. Rathod and H. Rao, Integration of trivariate polynomials over linear polyhedra in Euclidean three-dimensional space, J. Austral. Math. Soc. Ser., 39 (1998), 355–385.

[24] J.-C. Robinet, J. Gressier, G. Casalis and J.-M. Moschetta, Shock wave instability and the carbuncle phenomenon: same intrinsic origin, J. Fluid Mech., 417 (2000), 237–263.

[25] C.-W. Shu, High Order Finite Difference and Finite Volume WENO Schemes and Discontinuous Galerkin Methods for CFD, May 2001, NASA/CR-2001-210865.

[26] S. Tu, A high order space-time Riemann-solver-free method for solving compressible Euler equations, January 2009, AIAA Paper 2009–1335.

[27] S. Tu, A solution limiting procedure for an arbitrarily high order space-time method, June 2009, AIAA Paper 2009–3983.

[28] S. Tu and S. Aliabadi, A space-time upwind cell-vertex scheme for conservation laws: a Riemann solver-free approach, In K. Bathe, editor, Proceedings of the Third M.I.T. Conference on Computational Fluid and Solid Mechanics, pages 1191–1195, Elsevier Ltd., 2005.

[29] S. Tu and Z. Tian, Preliminary implementation of a high order space-time method on overset cartesian/quadrilateral grids, January 2010, AIAA Paper 2009-0544.

[30] J. van der Vegt and H. van der Ven, Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows. I. General formulation, J. Comput. Phys., 182 (2002), 546–585.

[31] Z. Wang, Spectral (finite) volume method for conservation laws on unstructured grids, J. Comput. Phys., 178 (2002), 210–251.