MATHEMATICS OF COMPUTATION
Volume 89, Number 324, July 2020, Pages 1807–1842
https://doi.org/10.1090/mcom/3507
Article electronically published on January 29, 2020

HIGH ORDER EXPLICIT LOCAL TIME STEPPING METHODS FOR HYPERBOLIC CONSERVATION LAWS

THI-THAO-PHUONG HOANG, LILI JU, WEI LENG, AND ZHU WANG

Abstract. In this paper we present and analyze a general framework for constructing high order explicit local time stepping (LTS) methods for hyperbolic conservation laws. In particular, we consider the model problem discretized by Runge-Kutta discontinuous Galerkin (RK-DG) methods and design LTS algorithms based on the strong stability preserving Runge-Kutta (SSP-RK) schemes, that allow spatially variable time step sizes to be used for time integration in different regions of the computational domain. The proposed algorithms are of predictor-corrector-type, in which the interface information along the time direction is first predicted based on the SSP-RK approximations and Taylor expansions, and then the fluxes over the region of the interface are corrected to conserve mass exactly at each time step. Following the proposed framework, we detail the corresponding LTS schemes with accuracy up to the fourth order, and prove their conservation property and nonlinear stability for the scalar conservation laws. Numerical experiments are also presented to demonstrate excellent performance of the proposed LTS algorithms.

1. Introduction

Numerical methods for hyperbolic conservation laws are a subject of great interest and importance as these laws are extensively used for modeling a wide range of physical phenomena such as gas dynamics, shallow water flows, advection of contaminants, traffic flows, etc. It is well known that these problems are often highly nonlinear and may develop discontinuous solutions with sharp and moving fronts/shocks. To obtain accurate and stable numerical solutions to hyperbolic conservation laws, it is popular to use conservative high resolution methods in space together with explicit time stepping. Examples of such spatial discretization include the MUSCL (monotonic upwind scheme for conservation laws) [56], the ENO (essentially nonoscillatory) and WENO (weighted ENO) schemes [25, 26, 29, 34], and the RK-DG (Runge-Kutta discontinuous Galerkin) methods [6–9]. Note that to guarantee numerical stability, the time step size needs to satisfy the CFL condition, which is determined by the spatial mesh size and wave speed. The use of local spatial refinements is efficient in resolving the sharp, moving fronts. However, as the CFL condition needs to hold everywhere, the step size for time integration would be controlled by the smallest cell size, or by the highest wave speed, which certainly increases the computational cost as a small time step size has to be used

Received by the editor May 22, 2019, and, in revised form, October 24, 2019.
2010 *Mathematics Subject Classification.* Primary 65M20, 65L06, 65M12.
This work was partially supported by the U.S. Department of Energy under grant numbers DE-SC0016540 and DE-SC0020270 and the U.S. National Science Foundation under grant numbers DMS-1818438 and DMS-1913073.

globally. Thus, to improve computational efficiency, the global CFL condition could be replaced by local ones so that different time step sizes can be used in different regions: smaller time step sizes where the mesh is fine or the wave speed is high, and larger time step sizes where the mesh is coarse or the wave speed is low.

Explicit local time stepping (LTS) algorithms have a long tradition. To the best of our knowledge, the first LTS algorithm for hyperbolic conservation laws was introduced in [39] for the one-dimensional scalar case based on the forward Euler method in time. It is of predictor-corrector-type and is first order accurate in both space and time. Extension to high resolution schemes with slope limiters for advection equations was presented in [12], and to second order in time for hyperbolic conservation laws in [13]. The numerical results on two-dimensional test problems confirm that these LTS schemes are very competitive to the global time stepping methods with respect to the accuracy in time. The application of LTS schemes to the shallow water equations was investigated in [43] with a Godunov-type finite volume discretization in space and later in [55] using the RK-DG finite element methods. Note that the LTS scheme in [43] is only first order accurate in time, while the one in [55] is second order accurate in time on regions away from the LTS interface but its accuracy degrades to first order at the interface. The LTS scheme in [55] is based on the second order strong stability preserving Runge-Kutta (SSP-RK) method, which is also known as a total variation diminishing (TVD) method introduced in [45,49]. Higher order RK-based explicit LTS methods were introduced for conservation laws in [1, 31] and for wave propagation in [21]. In [14], a space-time fully adaptive multiresolution method based on natural continuous extensions for RK methods was proposed, whose accuracy is of second order in both space and time. Local space-time refinement based on the discontinuous Galerkin (DG) approach was also extensively investigated in [15, 35, 52]. In particular, high order one step DG schemes based on ADER (arbitrary order using derivatives) [54] with local spatial and temporal refinement were studied for one-dimensional conservation laws in [35], for elastic waves with p-adaptivity in [15] and for Maxwell equations in [52] on unstructured tetrahedral meshes. In [16] a high order Lagrangian finite volume scheme on nonconforming space-time meshes for hyperbolic conservation laws was presented in which LTS was constructed based on high order accurate one step ADER time discretization. Other works related to LTS include the adaptive mesh refinement (AMR) method [2, 3, 18], the multirate time stepping method [11, 44] and the Implicit-Explicit (IMEX) based LTS methods [22, 42]. Among them, the AMR method involves the refinement in both space and time, i.e., small time step sizes are taken on the refined mesh and large time step sizes on the coarse mesh. It is different from our approach in the way that refined grids are placed over regions of the coarse grid and information is exchanged between the grids by means of injection and interpolation. The multirate time stepping method allows different time step sizes in different regions but it requires buffer regions to accommodate the time scale transition between regions. An overview of LTS techniques over the last two decades can be found in [19].

In [27], inspired by the first order predictor-corrector scheme in [39], we have designed conservative second and third order explicit LTS algorithms, incorporating with SSP-RK, for the rotating shallow water equations. The model is discretized in space by a C-grid staggering finite volume method, namely the TRiSK scheme [41, 53], on orthogonal primal and dual meshes. Numerical results with

parallel implementation demonstrate excellent performance of the LTS algorithms in terms of stability, accuracy, efficiency, and scalability. In this work, we extend the approach to construct, in a systematic way, a framework of high order LTS algorithms for hyperbolic conservative laws. In order to derive high order LTS algorithms, the key idea is to find high order approximations on the interface at intermediate time levels to handle the coupling between coarse and fine time steppings. Our proposed schemes are also of predictor-corrector-type: we derive the predictors based on Taylor series expansions of the solution at the current time level and the SSP-RK stepping algorithms at each intermediate time level. Our approach thus is different from the one proposed in [1, 31] where the predictors are based on RK time stepping and interpolating polynomials. We present up to fourth order predictors within this framework, and show that the proposed LTS schemes preserve the accuracy in time over the entire domain. Concerning the corrector, it is designed to balance the fluxes from the regions with small time step sizes to the ones with large time step sizes. As high order SSP-RK methods consist of multiple stages, the fluxes at the same stage are accumulated over all the intermediate time levels to update the interface solution associated with that stage. As a consequence, the total mass is well conserved, though the corrector is no longer convex combinations of forward Euler steps as in the global SSP-RK methods. Nevertheless, we rigorously prove that the proposed LTS schemes for scalar conservation laws are total variation bounded (TVB). Such nonlinear stability is a crucial feature of any effective numerical methods for hyperbolic conservation laws because it guarantees that the schemes can capture moving shocks without introducing nonphysical oscillations. Various numerical experiments are carried out to validate the accuracy, conservation, and stability of our LTS schemes. Since time advancement of the simulations in the fine regions and in the coarse ones can be implemented in parallel (this will be discussed further in Section 3), the proposed LTS schemes preserve the natural parallelism of explicit time stepping schemes.

Consider the initial value problem for hyperbolic conservation laws:

$$\begin{array}{ll} \frac{\partial \mathbf{u}}{\partial t} + \sum_{i=1}^{d} \frac{\partial \mathbf{f}_i}{\partial x_i}(\mathbf{u}) = \mathbf{0} & \text{in } \mathbb{R}^d \times (0,T), \\ \mathbf{u}(\mathbf{x},0) = \mathbf{u}_0(\mathbf{x}) & \text{in } \mathbb{R}^d, \end{array} \tag{1.1}$$

where $\mathbf{u}(\mathbf{x}) := (u_1(\mathbf{x}), \ldots, u_m(\mathbf{x}))$ is an $m-$dimensional vector of unknowns and each flux function $\mathbf{f}_i : \mathbb{R}^m \longrightarrow \mathbb{R}^m$ defined by

$$\mathbf{u} \quad \mapsto \quad \mathbf{f}_i(\mathbf{u}) := (f_{i1}(\mathbf{u}), \ldots, f_{im}(\mathbf{u}))$$

is vector-valued and is of m components. Since we focus on the time discretization techniques in this paper, we shall only consider the one-dimensional case, $d = 1$. In particular, our model problem is the following scalar hyperbolic conservation law, resulting from the system (1.1) with $d = m = 1$:

$$\begin{array}{ll} \frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 & \text{in } \mathbb{R} \times (0,T), \\ u(x,0) = u_0 & \text{in } \mathbb{R}. \end{array} \tag{1.2}$$

We shall construct and analyze high order Runge-Kutta discontinuous Galerkin algorithms with local time stepping for (1.2). The proposed LTS algorithms can be straightforwardly extended to the case of one-dimensional systems of conservation laws ($m > 1$), which will be presented in the numerical results, as well as to the higher-dimensional problems ($d > 1$).

The rest of this paper is structured as follows. In Section 2 we briefly introduce the RK-DG methods for scalar conservation laws (1.2). High order LTS algorithms are carefully derived in Section 3, and their conservation and stability properties are then proved in Section 4. Numerical results for various test cases are given in Section 5 to demonstrate the performance of the proposed LTS schemes. Additionally, coefficients of the SSP-RK methods used in the paper are given in Appendix A, and detailed derivation of the predictors for the proposed LTS schemes is presented in Appendix B.

2. Runge-Kutta discontinuous Galerkin methods

We first introduce the RK-DG methods and refer to [7] for a complete presentation of the methods. Within the framework of RK-DG, we first discretize equation (1.2) in space by the discontinuous Galerkin method, then integrate it in time by SSP-RK schemes, and finally apply a slope limiter to achieve stable and high order accurate numerical solutions.

2.1. **Spatial discretization by the discontinuous Galerkin.** Assume a partition of the real line $\mathbb{R}$ to have the jth intervals as $I_j = \left(x_{j-1/2}, x_{j+1/2}\right)$ and define $\Delta_j = x_{j+1/2} - x_{j-1/2}$ and $h = \max_j \Delta_j$. Let V_h be the finite-dimensional space consisting of discontinuous, piecewise polynomial functions:

$$V_h = V_h^k = \left\{v \in L^1(\mathbb{R}) : v\,|_{I_j} \in \mathcal{P}^k(I_j)\ \forall\, j\right\}\ \not\subset H^1(\mathbb{R}),$$

where $\mathcal{P}^k(I_j)$ is the space of polynomials of degree at most k on I_j. Consider a weak formulation of (1.2) obtained from testing it by any function $v_h \in V_h$ over I_j.

For a.e. $t \in (0,T)$, find $u_h(t) \in V_h$ such that: for all j and for all $v_h \in V_h$

$$\begin{aligned} &\textstyle\int_{I_j} \partial_t u_h(x,t)\, v_h(x)\, dx - \int_{I_j} f(u_h(x,t))\, \partial_x v_h(x)\, dx \\ &\qquad + h(u_h)_{j+1/2}(t)\, v_h(x^-_{j+1/2}) - h(u_h)_{j-1/2}(t)\, v_h(x^+_{j-1/2}) = 0, \\ &\textstyle\int_{I_j} u_h(x,0) v_h(x)\, dx = \int_{I_j} u_0(x) v_h(x)\, dx. \end{aligned} \tag{2.1}$$

Note that we have replaced the nonlinear flux $f(u(x_{j+1/2},t))$ in (2.1) by a *Lipschitz, consistent, monotone numerical flux* $h(u)_{j+1/2}(t)$ that depends on the two values of u at $x_{j+1/2}$:

$$h(u)_{j+1/2}(t) = h\left(u(x^-_{j+1/2},t), u(x^+_{j+1/2},t)\right).$$

The numerical flux $h(\cdot,\cdot)$ is required to satisfy the following properties: i) locally Lipschitz continuous; ii) consistent with the flux f, that is, $h(u,u) = f(u)$; and iii) nondecreasing in the first argument and nonincreasing in the second argument. Examples of such a flux include the Godunov flux, Engquist-Osher flux, Lax-Friedrichs flux, and Roe flux.

A local orthogonal basis of V_h consists of functions $\varphi_j^{(l)}$ defined as, for any j,

$$\varphi_j^{(l)} := P_l\left(\frac{2(x-x_j)}{\Delta_j}\right) \quad \text{for } l = 0,1,\ldots,k,$$

in which P_l is the Legendre polynomial of degree l and x_j is the middle point of I_j. Consequently, the approximate solution u_h is expressed uniquely as

$$u_h(x,t) = \sum_{l=0}^{k} u_j^{(l)}(t)\, \varphi_j^{(l)}(x) \quad \text{for } x \in I_j, \tag{2.2}$$

where the degrees of freedom $u_j^{(l)}(t)$ are determined by

$$u_j^{(l)}(t) := \frac{2l+1}{\Delta_j}\int_{I_j} u(x,t)\varphi_j^{(l)}(x)\,dx \quad \text{for } l = 0, 1, \ldots, k.$$

Note that $u_j^{(0)}$ is the cell average of u in I_j. By taking $v_h = \varphi_j^{(l)}$ in (2.1), we obtain the following ODE for $u_j^{(l)}$ for any j:

$$\begin{aligned}\left(\tfrac{1}{2l+1}\right)\tfrac{du_j^{(l)}(t)}{dt} - \tfrac{1}{\Delta_j}\textstyle\int_{I_j} f(u_h(x,t))\,\partial_x\varphi_j^{(l)}\,dx \\ + \tfrac{1}{\Delta_j}\left[h(u_h)_{j+1/2}(t) - (-1)^l h(u_h)_{j-1/2}(t)\right] = 0 \quad \forall\, l = 0, 1, \ldots, k,\end{aligned} \tag{2.3}$$

with the initial condition

$$u_j^{(l)}(0) = \frac{2l+1}{\Delta_j}\int_{I_j} u_0(x)\varphi_j^{(l)}(x)\,dx.$$

Note that in (2.3) we have used the following properties of Legendre polynomials:

$$\varphi_j^{(l)}(x_{j+1/2}^-) = P_l(1) = 1, \quad \varphi_j^{(l)}(x_{j-1/2}^+) = P_l(-1) = (-1)^l.$$

The numerical flux h is computed by $h(u_h)_{j+1/2}(t) = h\left(u_{j+1/2}^-(t), u_{j+1/2}^+(t)\right)$, where $u_{j+1/2}^{\pm}(t) = u_h(x_{j+1/2}^{\pm}, t)$ are defined as

$$u_{j+1/2}^-(t) = \sum_{l=0}^{k} u_j^{(l)}, \quad u_{j-1/2}^+(t) = \sum_{l=0}^{k}(-1)^l u_j^{(l)}.$$

Approximating the integral in (2.3) by Gauss-Lobatto quadrature rules that involve the two endpoints of the interval yields (using the definition of u_h in (2.2)):

$$\int_{I_j} f(u_h)\,\partial_x\varphi_j^{(l)}\,dx = \int_{I_j} f\left(u_{j-1/2}^+, (u_j^{(l)})_{0\le l\le k}, u_{j+1/2}^-\right)\,\partial_x\varphi_j^{(l)}\,dx.$$

The system of ODEs (2.3) can be recast in an autonomous form as follows:

$$\frac{d\mathbf{U}_h}{dt} = \mathbf{L}_h(\mathbf{U}_h), \quad \mathbf{U}_h(0) = \mathbf{U}_{h0}, \tag{2.4}$$

where $\mathbf{U}_h = (\mathbf{u}_j)_{\forall\, j}$ with $\mathbf{u}_j = (u_j^{(l)})_{l=0,\ldots,k}$, the right-hand side

$$\mathbf{L}_h(\mathbf{U}_h) = \left(L_{h,j}^{(l)}(u_{j-1/2}^{\pm}, \mathbf{u}_j, u_{j+1/2}^{\pm})\right)_{\forall\, j,\, l=0,1,\ldots,k}$$

with

$$\begin{aligned}L_{h,j}^{(l)}(u_{j-1/2}^{\pm}, \mathbf{u}_j, u_{j+1/2}^{\pm}) = \tfrac{2l+1}{\Delta_j}\Big\{ \textstyle\int_{I_j} f\left(u_{j-1/2}^+, \mathbf{u}_j, u_{j+1/2}^-\right)\,\partial_x\varphi_j^{(l)}\,dx \\ - \left[h(u_{j+1/2}^-, u_{j+1/2}^+) - (-1)^l h(u_{j-1/2}^-, u_{j-1/2}^+)\right]\Big\},\end{aligned} \tag{2.5}$$

and the initial data $\mathbf{U}_{h0} = \left[{}^{(2l+1)}/_{\Delta_j}\int_{I_j} u_0(x)\varphi_j^{(l)}(x)\,dx\right]_{\forall\, j,\, l=0,1,\ldots,k}$. Next, we solve (2.4) explicitly in time by the SSP-RK methods [20, 48].

2.2. **Strong stability preserving Runge-Kutta time discretization.** The SSP-RK methods have been proved to be effective for solving hyperbolic conservation laws with discontinuous solutions. Given a uniform partition of $(0,T)$, $0 = t_0 < t_1 < \ldots < t_{N-1} < t_N = T$ with the time step size $\Delta t = T/N$. The s-stage, rth-order SSP-RK methods, referred to as SSP-RK(s,r), for solving the autonomous system (2.4) read as follows: for $n = 0, \ldots, N-1$, compute

$$\mathbf{U}_h^{n,(i)} = \sum_{\nu=0}^{i-1} \alpha_{i\nu} \mathbf{U}_h^{n,(\nu)} + \beta_{i\nu} \Delta t \mathbf{L}_h(\mathbf{U}_h^{n,(\nu)}) \quad \forall\, i = 1, \ldots, s, \tag{2.6}$$

where $\mathbf{U}_h^{n,(0)} = \mathbf{U}_h^n$, and set $\mathbf{U}_h^{n+1} = \mathbf{U}_h^{n,(s)}$. It is required that all the weights $\alpha_{i\nu}, \beta_{i\nu} \geq 0$. To measure stability of RK-DG methods, we denote by $\mathbf{u}^{(l),n} = (u_j^{(l),n})_{\forall j}$ and define the total variation of numerical solutions by

$$TV(\mathbf{u}^{(l),n}) = \sum_j \left| u_{j+1}^{(l),n} - u_j^{(l),n} \right| \quad \forall\, l = 0, \ldots, k \text{ and } \quad n = 0, \ldots, N-1.$$

A numerical method is *total variation diminishing* (TVD) if

$$TV(\mathbf{u}^{(l),n+1}) \leq TV(\mathbf{u}^{(l),n}) \quad \forall\, l = 0, \ldots, k \text{ and } \quad n = 0, \ldots, N-1,$$

and is *total variation bounded* (TVB) if

$$TV(\mathbf{u}^{(l),n+1}) \leq TV(\mathbf{u}^{(l),0}) + BT \quad \forall\, l = 0, \ldots, k \text{ and } \quad n = 0, \ldots, N-1$$

for some constant B independent of the time step size. The stability of the SSP-RK schemes is given by the following lemma.

Lemma 2.1 ([20]). *If the forward Euler method* $\mathbf{U}_h^{n+1} = \mathbf{U}_h^n + \Delta t \mathbf{L}_h(\mathbf{U}_h^n)$ *is TVD under the CFL condition* $\Delta t \leq \Delta t_{FE}$*, then the SSP-RK*$(s,r)$ *scheme* (2.6) *is TVD under the modified CFL condition:* $\Delta t \leq \mathcal{C} \Delta t_{FE}$*, where* $\mathcal{C} := \min_{i,\nu} \frac{\alpha_{i\nu}}{\beta_{i\nu}}$ *is the SSP coefficient.*

We present some commonly used SSP-RK schemes such as SSP-RK(2,2), SSP-RK(3,3), and SSP-RK(5,4) in detail in Appendix A.

2.3. **TVB corrected slope limiter.** In order to handle moving shocks while preserving high order accuracy in smooth regions, we follow [45] and define the TVB corrected *minmod* function $\widetilde{m}$:

$$\widetilde{m}(a_1, \ldots, a_\nu) = \begin{cases} a_1 & \text{if } |a_1| \leq C_M h^2, \\ m(a_1, \ldots, a_\nu) & \text{otherwise,} \end{cases}$$

where $C_M > 0$ is a constant and m is the usual *minmod* function [23]:

$$m(a_1, \ldots, a_\nu) = s \min_{1 \leq i \leq \nu} |a_i| \quad \text{with } s = \begin{cases} \operatorname{sign}(a_1) & \text{if } \operatorname{sign}(a_1) = \ldots = \operatorname{sign}(a_\nu), \\ 0 & \text{otherwise.} \end{cases} \tag{2.7}$$

The corrected limiter leads to high order accuracy in any region where the solution is smooth, even at local extrema. The resulting scheme is no longer TVD, instead it is TVB. Next, we define the $(k+1)$th-order limiter $\Lambda\Pi_h^k$ as in [5]. When $k = 1$, we have

$$\Lambda\Pi_h^1(u_h)|_{I_j} = u_j^{(0)} + \widetilde{m}(u_j^{(1)}, u_{j+1}^{(0)} - u_j^{(0)}, u_j^{(0)} - u_{j-1}^{(0)}) \varphi_j^{(1)}(x).$$

For $k > 1$, we first compute

$$\begin{aligned} u_{j+1/2}^{-(\mathrm{mod})} &= u_j^{(0)} + \widetilde{m}(u_{j+1/2}^- - u_j^{(0)}, u_{j+1}^{(0)} - u_j^{(0)}, u_j^{(0)} - u_{j-1}^{(0)}), \\ u_{j-1/2}^{+(\mathrm{mod})} &= u_j^{(0)} - \widetilde{m}(u_j^{(0)} - u_{j-1/2}^+, u_{j+1}^{(0)} - u_j^{(0)}, u_j^{(0)} - u_{j-1}^{(0)}), \end{aligned}$$

then define

$$\Lambda\Pi_h^k(u_h)|_{I_j} = \begin{cases} u_h|_{I_j} & \text{if } u_{j+1/2}^{-(\mathrm{mod})} = u_{j+1/2}^- \text{ and } u_{j-1/2}^{+(\mathrm{mod})} = u_{j-1/2}^+, \\ \Lambda\Pi_h^1(u_h)|_{I_j} & \text{otherwise.} \end{cases}$$

We finally introduce the following notation

$$u_h^{(\mathrm{mod})}|_{I_j} := \Lambda\Pi_h^k(u_h)|_{I_j} = \sum_{l=0}^{k} u_j^{(l)(\mathrm{mod})} \varphi_j^{(l)}$$

and

$$\Lambda\Pi_h^k(\mathbf{U}_h) := \mathbf{U}_h^{(\mathrm{mod})} = \left[u_j^{(l)(\mathrm{mod})} \right]_{\forall\, j\, \forall\, l=0,1,\ldots,k}.$$

The complete RK-DG method with the TVB *minmod* limiter is given in Algorithm 1, in which $r = (k+1)$ to match the accuracy in space and in time, and $s \geq r$ is the number of stages in SSP-RK.

Algorithm 1 Runge-Kutta local projection discontinuous Galerkin method

1: Compute $\mathbf{U}_h^{0(\mathrm{mod})} = \Lambda\Pi_h^k(\mathbf{U}_{h0})$.
2: For each $n = 0, 1, \ldots, N-1$,
   (1) Set $\mathbf{U}_h^{n,(0)(\mathrm{mod})} = \mathbf{U}_h^{n(\mathrm{mod})}$.
   (2) For $i = 1, \ldots, s$, compute the solution at stage i:

$$\mathbf{U}_h^{n,(i)(\mathrm{mod})} = \Lambda\Pi_h^k \left(\sum_{\nu=0}^{i-1} \alpha_{i\nu} \mathbf{U}_h^{n,(\nu)(\mathrm{mod})} + \beta_{i\nu} \Delta t \mathbf{L}_h \left(\mathbf{U}_h^{n,(\nu)(\mathrm{mod})} \right) \right).$$

   (3) Set $\mathbf{U}_h^{n+1(\mathrm{mod})} = \mathbf{U}_h^{n,(s)(\mathrm{mod})}$.

3. Local time-stepping algorithms

In this section, we present high order LTS algorithms incorporated with the RK-DG methods for conservation laws. Given the solution $\mathbf{U}_h^{n(\mathrm{mod})}$ at t^n, possibly with moving shocks, we approximate the solution at t^{n+1}. To this end, we divide the domain into coarse and fine regions, and assume shocks only appear in the fine regions. This could be made possible by varying the LTS interfaces with time. Consequently, we can use spatially variable time steps: large step sizes in the coarse regions and small step sizes in the fine regions.

For simplicity of presentation, we decompose the domain into a coarse region Ω_c^n and a fine region Ω_f^n. Extension to more complicated configurations with multiple subdomains is straightforward. Denoted by $x_{j_0^n+1/2}$ the interface point at t^n, $\Omega_c^n = \{I_j : j \leq j_0^n\}$ the coarse region, and $\Omega_f^n = \{I_j : j \geq j_0^n + 1\}$ the fine region. As depicted in Figure 1, we enforce a larger time step size $\Delta t_{\mathrm{coarse}} = \Delta t$ in Ω_c^n and

a smaller time step size $\Delta t_{\text{fine}} = {}^{\Delta t}/_{M}$ in Ω_f^n. We remark that the coarse time increment must be a union of fine time increments:

$$\left[t^n, t^{n+1}\right) = \bigcup_{p=0}^{M-1} \left[t^{n,p}, t^{n,p+1}\right).$$

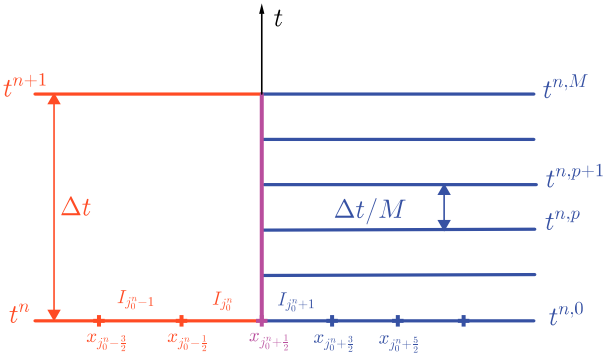


FIGURE 1. Partition in time with local time stepping.

To proceed in time in the fine region, one needs to find $(k+1)$th-order in time approximation of the flux at the interface at intermediate time levels $t^{n,p}$ for $p = 1, \ldots, M-1$. This is obtained via a predictor based on kth-order Taylor expansions and the $(k+1)$th-order SSP-RK algorithm, assuming that the solution is smooth enough near the LTS interface. After advancing in the fine region to t^{n+1}, we will correct the flux at the interface in order to conserve mass exactly. The derivation of the predictors up to fourth order accuracy is presented in Appendix B. The proposed LTS algorithm of order $(k+1)$ consists of the following three steps.

Step 1 (Predicting the interface values). We first compute the solution of the first $(s-1)$ stages of the SSP-RK$(s, k+1)$ scheme on the interface cell $I_{j_0^n}$ with the coarse time step size:

$$\mathbf{u}_{j_0^n}^{n,(i)} = \left(u_{j_0^n}^{(l),n,(i)}\right)_{\forall\, j,\, l=0,\ldots,k} \qquad \forall\, i = 1, \ldots, s-1.$$

It is important to note that we compute $\mathbf{u}_{j_0^n}^{n,(i)}$ *locally* on I_{j_0} by enforcing $u_{j_0^n-1/2}^{n,(i),-} = u_{j_0^n-1/2}^{n,(i),+}$ and $u_{j_0^n+1/2}^{n,(i),+} = u_{j_0^n+1/2}^{n,(i),-}$ in (2.5). This is obtained under the assumption that the solution near the LTS interface is continuous (for $k \le 1$) or sufficiently smooth (for $k > 1$). Thus, applying the limiter is not necessary in this case and we have $\mathbf{u}_{j_0^n}^{n,(i)(\text{mod})} = \mathbf{u}_{j_0^n}^{n,(i)}$. We then use these values to predict the solution on the interface $x_{j_0^n+1/2}$ at intermediate time levels $t^{n,p}$:

$$u_{j_0^n+1/2}^{n,p,(i),-(\text{mod})} = u_{j_0^n+1/2}^{n,p,(i),-} = \sum_{l=0}^{k} u_{j_0^n}^{(l),n,p,(i)} \quad \forall\, p = 0, 1, \ldots, M-1, \tag{3.1}$$

where $u_{j_0^n}^{(l),n,p,(i)}$ are computed by the formulas in Appendix B. In particular:

For the second order SSP-RK(2,2):

$$\begin{aligned} u_{j_0^n}^{(l),n,p,(0)} &= (1-\theta_p)u_{j_0^n}^{(l),n,(0)} + \theta_p u_{j_0^n}^{(l),n,(1)}, \\ u_{j_0^n}^{(l),n,p,(1)} &= (1-\eta_p)u_{j_0^n}^{(l),n,(0)} + \eta_p u_{j_0^n}^{(l),n,(1)}, \end{aligned} \tag{3.2}$$

for $l = 0, 1$, where $\theta_p = \frac{p}{M}$ and $\eta_p = \frac{p+1}{M}$ for $p = 0, 1, \ldots, M-1$.

For the third order SSP-RK(3,3):

$$\begin{aligned} u_{j_0^n}^{(l),n,p,(0)} &= (1-\theta_p-\widehat{\theta}_p)u_{j_0^n}^{(l),n,(0)} + (\theta_p-\widehat{\theta}_p)u_{j_0^n}^{(l),n,(1)} + 2\widehat{\theta}_p u_{j_0^n}^{(l),n,(2)}, \\ u_{j_0^n}^{(l),n,p,(1)} &= (1-\eta_p-\widehat{\eta}_p)u_{j_0^n}^{(l),n,(0)} + (\eta_p-\widehat{\eta}_p)u_{j_0^n}^{(l),n,(1)} + 2\widehat{\eta}_p u_{j_0^n}^{(l),n,(2)}, \\ u_{j_0^n}^{(l),n,p,(2)} &= (1-\gamma_p-\widehat{\gamma}_p)u_{j_0^n}^{(l),n,(0)} + (\gamma_p-\widehat{\gamma}_p)u_{j_0^n}^{(l),n,(1)} + 2\widehat{\gamma}_p u_{j_0^n}^{(l),n,(2)}, \end{aligned} \tag{3.3}$$

for $l = 0, 1, 2$, with θ_p and η_p as above, and $\widehat{\theta}_p = \frac{p^2}{M^2}$, $\widehat{\eta}_p = \frac{p(p+2)}{M^2}$, $\gamma_p = \frac{2p+1}{2M}$, and $\widehat{\gamma}_p = \frac{2p^2+2p+1}{2M^2}$ for $p = 0, 1, \ldots, M-1$.

For the fourth order SSP-RK(5,4): we approximate $u_{j_0^n}^{(l),n,p,(i)}$, for $p = 0, 1, \ldots, M-1$ as linear combinations of $u_{j_0^n}^{(l),n,(i)}$ for $i = 0, \ldots, 4$, as presented in Appendix B.3.

Step 2 (Advancing in the coarse and fine regions in parallel).

Step 2a (Advancing the coarse region excluding the interface cell). With the solution at the current time level, we advance the solution to the next time level by running the SSP-RK with the coarse time step size.

For all the cells I_j with $j < j_0^n$, we perform:

(1) For $i = 1, \ldots, s$,

$$u_j^{(l),n,(i)} = \textstyle\sum_{\nu=0}^{i-1} \alpha_{i\nu} u_j^{(l),n,(\nu)} + \beta_{i\nu}\Delta t\, L_{h,j}^{(l)}\left(u_{j-1/2}^{n,(\nu),\pm}, \mathbf{u}_j^{n,(\nu)}, u_{j+1/2}^{n,(\nu),\pm}\right). \tag{3.4}$$

(2) Set $\mathbf{u}_j^{n+1} = \mathbf{u}_j^{n,(s)}$ for all $j < j_0^n$.

Step 2b (Advancing in the fine region). With the predicted values on the interface, we evaluate the interface flux $h(u_{j_0^n+1/2}^{n,p,-}, u_{j_0^n+1/2}^{n,p,+})$ at the intermediate time levels, and consequently obtain the solution $u_j^{n,p}$ for all the cells I_j with $j > j_0^n$ in the fine region. The TVB limiter is performed to obtain $u_j^{n,p(\mathrm{mod})}$ for $j > j_0^n$ and the predicted values are updated on the interface after limiting.

For all the cells I_j with $j \geq j_0^n + 1$, we perform:

For $p = 0, \ldots, M-1$,

(1) Set $u_j^{(l),n,p,(0)(\mathrm{mod})} = u_j^{(l),n,p(\mathrm{mod})}$ for $l = 0, \ldots, k$.

(2) For $i = 1, \ldots, s$, we compute the solution at stage i:

$$\begin{aligned} u_j^{(l),n,p,(i)} = {} & \textstyle\sum_{\nu=0}^{i-1} \alpha_{i\nu} u_j^{(l),n,p,(\nu)(\mathrm{mod})} \\ & + \beta_{i\nu}\left(\tfrac{\Delta t}{M}\right) L_{h,j}^{(l)}\left(u_{j-1/2}^{n,p,(\nu),\pm(\mathrm{mod})}, \mathbf{u}_j^{n,p,(\nu)(\mathrm{mod})}, u_{j+1/2}^{n,p,(\nu),\pm(\mathrm{mod})}\right) \end{aligned}$$

for $l = 0, \ldots, k$. If $p < M-1$, limit the solution in the fine region

$$\mathbf{u}_j^{n,p,(i)(\mathrm{mod})} = \Lambda\Pi_h^k\left(\mathbf{u}_{j' \geq j_0^n}^{n,p,(i)}\right)|_{I_j},$$

and update the predicted interface value $u_{j_0^n+1/2}^{n,p,(i),-(\mathrm{mod})}$ (cf. (3.1)) after limiting:

$$u_{j_0^n+1/2}^{n,p,(i),-(\mathrm{mod})} = u_{j_0^n}^{(0),n,p,(i)} + \widetilde{m}\Big(u_{j_0^n+1/2}^{n,p,(i),-(\mathrm{mod})} - u_{j_0^n}^{(0),n,p,(i)}, u_{j_0^n+1}^{(0),n,p,(i)} - u_{j_0^n}^{(0),n,p,(i)}, u_{j_0^n}^{(0),n,p,(i)} - u_{j_0^n-1}^{(0),n,(i)}\Big).$$

(3) For all $j > j_0^n$, set:

$$\begin{cases} u_j^{(l),n,p+1(\mathrm{mod})} = u_j^{(l),n,p,(s)(\mathrm{mod})} & \text{if } p < M-1, \\ u_j^{(l),n+1} = u_j^{(l),n,p,(s)} & \text{if } p = M-1. \end{cases}$$

Step 3 (Correcting the interface solution and limiting the global solution at t^{n+1} locally). With the predicted interface value $u_{j_0^n+1/2}^{n,p,(\nu),-}$, we calculate the flux at the interface $x = x_{j_0^n+1/2}$. Together with the flux at $x = x_{j_0^n-1/2}$, which is frozen over $[t^n, t^{n+1})$, we correct the solution of the interface cell $I_{j_0^n}$. Finally, the TVB limiter is applied, which can be implemented in parallel as [10], to limit the solution on I_j for all j in which only information on elements sharing edges with I_j is necessary.

(1) For $i = 1, \dots, s$, we compute the solution at stage i at the interface:

$$\begin{aligned} \widehat{u}_{j_0^n}^{(l),n,(i)} = {} & \textstyle\sum_{\nu=0}^{i-1} \alpha_{i\nu} \widehat{u}_{j_0^n}^{(l),n,(\nu)} \\ & + \beta_{i\nu} \tfrac{\Delta t}{M} \textstyle\sum_{p=0}^{M-1} L_{h,j_0^n}^{(l)} \left(u_{j_0^n-1/2}^{n,(\nu),\pm(\mathrm{mod})}, \mathbf{u}_{j_0^n}^{n,(\nu)(\mathrm{mod})}, u_{j_0^n+1/2}^{n,p,(\nu),\pm(\mathrm{mod})}\right), \end{aligned} \tag{3.5}$$

where $\widehat{u}_{j_0^n}^{(l),n,(0)} = u_{j_0^n}^{(l),n,(0)(\mathrm{mod})}$.

(2) Set $u_{j_0^n}^{(l),n+1} = \widehat{u}_{j_0^n}^{(l),n,(s)}$ and perform the limiter: $\mathbf{U}_h^{n+1(\mathrm{mod})} = \Lambda\Pi_h^k(\mathbf{U}_h^{n+1})$.

Remark 3.1. The proposed LTS algorithms can be naturally extended to the case in which more than two types of time step sizes are used in the entire domain, provided the larger time step size in any two adjacent subdomains is an integer multiple of the smaller one. A common procedure for implementing the predictor-corrector-type LTS algorithms with multiple time step sizes can be applied: common global time levels are set corresponding to the coarsest time step; local time levels are synchronized as needed; and the LTS interface solutions between the subdomains will be corrected sequentially, from the smallest time level to the global time level. A detailed description of such a procedure can be found in [31, Section 4.5], though the predictor and corrector in our algorithm are defined in this section. Numerical tests with multiple time step sizes will be presented in Section 5.

4. Properties of LTS schemes

First, we notice that the proposed LTS schemes preserve the accuracy in time of the corresponding global SSP-RK methods due to the construction of the predictor and the corrector (see also Remark B.1). In the following, we prove that the LTS schemes conserve mass exactly, and importantly, they satisfy the TVB stability.

4.1. **Conservation.** Mass conservation of the proposed LTS schemes is obtained via the construction of the corrector. For simplicity, we assume that the solutions are obtained after performing the limiter defined in Subsection 2.3 and write $u_j^{(l),n}$ for $u_j^{(l),n(\mathrm{mod})}$.

Theorem 4.1. *The LTS schemes exhibit exact conservation of mass:*

$$\int_{\mathbb{R}} u_h^{n+1} = \int_{\mathbb{R}} u_h^n \quad \forall\, n = 0, \dots, N-1.$$

Proof. We only need to show that mass is conserved in the region of the LTS interface $x = x_{j_0^n+1/2}$, $I_{j_0^n} \cup I_{j_0^n+1}$, under the assumption that no flux is imposed at $x_{j_0^n-1/2}$ and $x_{j_0^n+3/2}$:

$$\int_{I_{j_0^n} \cup I_{j_0^n+1}} u_h^{n+1} = \int_{I_{j_0^n} \cup I_{j_0^n+1}} u_h^n. \tag{4.1}$$

Next, we prove (4.1) for the second order LTS scheme based on SSP-RK(2, 2) (cf. Equations (A.1)). The proof for the third and fourth order LTS schemes can be done in a similar manner; in fact, the result holds for any high order LTS schemes with the corrector defined by (3.5).

For the fine cell $I_{j_0^n+1}$, the second order LTS algorithm reads:

$$\begin{aligned} u_{j_0^n+1}^{(l),n+1} &= \tfrac{1}{2} u_{j_0^n+1}^{(l),n,M-1} + \tfrac{1}{2} \Bigg[u_{j_0^n+1}^{(l),n,M-1,(1)} \\ &\qquad + \left(\tfrac{\Delta t}{M}\right) L_{h,j_0^n+1}^{(l)} \left(u_{j_0^n+1/2}^{n,M-1,(1),\pm}, \mathbf{u}_{j_0^n+1}^{n,M-1,(1)}, u_{j_0^n+3/2}^{n,M-1,(1),\pm} \right) \Bigg] \\ &= u_{j_0^n+1}^{(l),n,M-1} + \tfrac{1}{2} \tfrac{\Delta t}{M} \textstyle\sum_{\nu=0}^{1} L_{h,j_0^n+1}^{(l)} \left(u_{j_0^n+1/2}^{n,M-1,(\nu),\pm}, \mathbf{u}_{j_0^n+1}^{n,M-1,(\nu)}, u_{j_0^n+3/2}^{n,M-1,(\nu),\pm} \right). \end{aligned}$$

Thus, by recursion, we obtain:

$$u_{j_0^n+1}^{(l),n+1} = u_{j_0^n+1}^{(l),n} + \frac{1}{2} \frac{\Delta t}{M} \sum_{p=0}^{M-1} \sum_{\nu=0}^{1} L_{h,j_0^n+1}^{(l)} \left(u_{j_0^n+1/2}^{n,p,(\nu),\pm}, \mathbf{u}_{j_0^n+1}^{n,p,(\nu)}, u_{j_0^n+3/2}^{n,p,(\nu),\pm} \right). \tag{4.2}$$

Taking $v_h = 1$ in (2.1), using (4.2) and the definition of $L_{h,j}^{(l)}$ in (2.5), we have

$$\int_{I_{j_0^n+1}} u_h^{n+1} = \int_{I_{j_0^n+1}} u_h^n + \frac{1}{2} \frac{\Delta t}{M} \sum_{p=0}^{M-1} \left(-h(u_{j_0^n+1/2}^{n,p,-}, u_{j_0^n+1/2}^{n,p,+}) - h(u_{j_0^n+1/2}^{n,p,(1),-}, u_{j_0^n+1/2}^{n,p,(1),+}) \right) \tag{4.3}$$

as no flux is imposed at $x_{j_0^n+3/2}$.

For the interface cell $I_{j_0^n}$, the corrector (3.5) associated with SSP-RK(2, 2) is given by

$$\begin{aligned} \widehat{u}_{j_0^n}^{(l),n,(1)} &= u_{j_0^n}^{(l),n} + \tfrac{\Delta t}{M} \textstyle\sum_{p=0}^{M-1} L_{h,j_0^n}^{(l)} \left(u_{j_0^n-1/2}^{n,\pm}, \mathbf{u}_{j_0^n}^{n}, u_{j_0^n+1/2}^{n,p,\pm} \right), \\ u_{j_0^n}^{(l),n+1} &= \tfrac{1}{2} u_{j_0^n}^{(l),n} + \tfrac{1}{2} \left[\widehat{u}_{j_0^n}^{(l),n,(1)} + \tfrac{\Delta t}{M} \textstyle\sum_{p=0}^{M-1} L_{h,j_0^n}^{(l)} \left(u_{j_0^n-1/2}^{n,(1),\pm}, \mathbf{u}_{j_0^n}^{n,(1)}, u_{j_0^n+1/2}^{n,p,(1),\pm} \right) \right], \end{aligned}$$

from which we deduce that

$$u_{j_0^n}^{(l),n+1} = u_{j_0^n}^{(l),n} + \frac{1}{2} \frac{\Delta t}{M} \sum_{p=0}^{M-1} \sum_{\nu=0}^{1} L_{h,j_0^n}^{(l)} \left(u_{j_0^n-1/2}^{n,(\nu),\pm}, \mathbf{u}_{j_0^n}^{n,(\nu)}, u_{j_0^n+1/2}^{n,p,(\nu),\pm} \right). \tag{4.4}$$

As for the fine cell $j = j_0^n + 1$, we choose $v_h = 1$ in (2.1) and use (4.4) to obtain

$$\int_{I_{j_0^n}} u_h^{n+1} = \int_{I_{j_0^n}} u_h^n + \frac{1}{2} \frac{\Delta t}{M} \sum_{p=0}^{M-1} \left(h(u_{j_0^n+1/2}^{n,p,-}, u_{j_0^n+1/2}^{n,p,+}) + h(u_{j_0^n+1/2}^{n,p,(1),-}, u_{j_0^n+1/2}^{n,p,(1),+}) \right), \tag{4.5}$$

noting that no flux at $x_{j_0^n-1/2}$ is assumed. Thus, the proof is completed by adding (4.3) and (4.5) together. ∎

4.2. **Stability.** Numerical methods for conservation laws need to satisfy certain nonlinear stability requirements in order to prevent spurious oscillations when the solution is discontinuous. In [39], the first order LTS scheme based on forward Euler is proved to be TVD with the predictor obtained by freezing the value at t^n:

$$u^{n,p}_{j_0^n+1/2} = u^n_{j_0^n+1/2} \quad \forall p = 0, \dots, M-1.$$

For higher order LTS schemes as proposed in Section 3, multiple stage time stepping algorithms are employed and the predictors are obtained by taking linear combinations of the interface solution at different stages with the coarse time step size. Therefore, the proof of nonlinear stability for high order LTS schemes is not an obvious generalization from the first order one. Additionally, the corrector designed to conserve mass is not a convex combination of forward Euler steps as in the case of the global SSP-RK. As a consequence, the high order LTS schemes are not TVD anymore, instead they are TVB.

We next prove the stability of the second order LTS scheme by first showing that it is TVBM (total variation bounded in the means). The generalization to higher order LTS schemes can be done in a similar manner. We introduce some notation to be used in the proof. We denote by Δ_+ and Δ_- the forward and backward finite difference operators, respectively,

$$\Delta_+ u_j = u_{j+1} - u_j \text{ and } \quad \Delta_- u_j = u_j - u_{j-1}.$$

Following [7], we decompose the interface values $u^{\pm}_{j+1/2}$ as

$$u^-_{j+1/2} = \overline{u}_j + \widetilde{u}_j, \quad u^+_{j-1/2} = \overline{u}_j - \widetilde{\widetilde{u}}_j,$$

where $\overline{u}_j := u_j^{(0)}$ is the mean value of u on the cell I_j. As in [6], we denote:

$$C_{j+1/2} = -h_2 \cdot \left(1 - \frac{\Delta_+ \widetilde{\widetilde{u}}_j}{\Delta_+ \overline{u}_j}\right) \text{ and } \quad D_{j-1/2} = h_1 \cdot \left(1 + \frac{\Delta_- \widetilde{u}_j}{\Delta_- \overline{u}_j}\right), \tag{4.6}$$

where

$$h_1 = \frac{h(u^-_{j+1/2}, u^+_{j-1/2}) - h(u^-_{j-1/2}, u^+_{j-1/2})}{u^-_{j+1/2} - u^-_{j-1/2}}, \; h_2 = \frac{h(u^-_{j+1/2}, u^+_{j+1/2}) - h(u^-_{j+1/2}, u^+_{j-1/2})}{u^+_{j+1/2} - u^+_{j-1/2}}.$$

Note that h_1 and $-h_2$ are nonnegative due to the monotonicity of $h(\cdot, \cdot)$. Then the flux associated with the mean value $\overline{u}_j$ (cf. equation (2.5) with $l = 0$) can be rewritten equivalently as

$$-\Big(h(u^-_{j+1/2}, u^+_{j+1/2}) - h(u^-_{j-1/2}, u^+_{j-1/2})\Big) = C_{j+1/2}\, \Delta_+ \overline{u}_j - D_{j-1/2}\, \Delta_- \overline{u}_j.$$

Using the above notation, the second order LTS scheme as presented in Section 3 for the mean value $\overline{u}_j$ reads as follows: for $n = 0, \dots, N-1$,

(1) Compute the predicted mean on the interface cell at the intermediate time levels from the solutions with the coarse time step size: for $p = 0, \dots, M-1$,

$$\begin{aligned} \overline{u}^{n,p}_{j_0^n} &= \left(1 - \tfrac{p}{M}\right) \overline{u}^n_{j_0^n} + \tfrac{p}{M} \overline{u}^{n,(1)}_{j_0^n+1/2}, \\ \overline{u}^{n,p,(1)}_{j_0^n} &= \left(1 - \tfrac{p+1}{M}\right) \overline{u}^n_{j_0^n} + \tfrac{p+1}{M} \overline{u}^{n,(1)}_{j_0^n+1/2}. \end{aligned} \tag{4.7}$$

(2) Advance in the coarse region for all $j < j_0^n$:

$$\begin{aligned}\overline{u}_j^{n,(1)} &= \overline{u}_j^n + \tfrac{\Delta t}{\Delta x_j}\left(C_{j+1/2}^n \Delta_+ \overline{u}_j^n - D_{j-1/2}^n \Delta_- \overline{u}_j^n\right),\\ \overline{u}_j^{n+1} &= \tfrac{1}{2}\overline{u}_j^n + \tfrac{1}{2}\left[\overline{u}_j^{n,(1)} + \tfrac{\Delta t}{\Delta x_j}\left(C_{j+1/2}^{n,(1)} \Delta_+ \overline{u}_j^{n,(1)} - D_{j-1/2}^{n,(1)} \Delta_- \overline{u}_j^{n,(1)}\right)\right],\end{aligned} \tag{4.8}$$

and in the fine region, for all $j > j_0^n$, for $p = 0, \dots, M-1$,

$$\begin{aligned}\overline{u}_j^{n,p,(1)} &= \overline{u}_j^{n,p} + \tfrac{\Delta t}{M\Delta x_j}\left(C_{j+1/2}^{n,p} \Delta_+ \overline{u}_j^{n,p} - D_{j-1/2}^{n,p} \Delta_- \overline{u}_j^{n,p}\right),\\ \overline{u}_j^{n,p+1} &= \tfrac{1}{2}\overline{u}_j^{n,p} + \tfrac{1}{2}\left[\overline{u}_j^{n,p,(1)} + \tfrac{\Delta t}{M\Delta x_j}\left(C_{j+1/2}^{n,p,(1)} \Delta_+ \overline{u}_j^{n,p,(1)} - D_{j-1/2}^{n,p,(1)} \Delta_- \overline{u}_j^{n,p,(1)}\right)\right].\end{aligned} \tag{4.9}$$

Note that the interface values $u_{j_0^n+1/2}^{n,p,(i),-}$, for $i = 1, 2$, and $p = 0, \dots, M-1$ are computed by using the second order predictor (4.7) and (3.1).

(3) Correcting the interface values for which the flux at $x = x_{j_0^n - 1/2}$ is frozen over $[t^n, t^{n+1})$:

$$\begin{aligned}\widehat{\overline{u}}_{j_0^n}^{n,(1)} &= \overline{u}_{j_0^n}^n - \tfrac{\Delta t}{M\Delta x_{j_0^n}} \textstyle\sum_{p=0}^{M-1}\left(h(u_{j_0^n+1/2}^{n,p,-}, u_{j_0^n+1/2}^{n,p,+}) - h(u_{j_0^n-1/2}^{n,-}, u_{j_0^n-1/2}^{n,+})\right),\\ \overline{u}_{j_0^n}^{n+1} &= \tfrac{1}{2}\overline{u}_{j_0^n}^n + \tfrac{1}{2}\left[\widehat{\overline{u}}_{j_0^n}^{n,(1)} - \tfrac{\Delta t}{M\Delta x_{j_0^n}} \textstyle\sum_{p=0}^{M-1}\left(h(u_{j_0^n+1/2}^{n,p,(1),-}, u_{j_0^n+1/2}^{n,p,(1),+}) - h(u_{j_0^n-1/2}^{n,(1),-}, u_{j_0^n-1/2}^{n,(1),+})\right)\right].\end{aligned} \tag{4.10}$$

The flux term in the right-hand side of (4.10) can be rewritten as

$$\begin{aligned}-\left(h(u_{j_0^n+1/2}^{n,p,-}, u_{j_0^n+1/2}^{n,p,+}) - h(u_{j_0^n-1/2}^{n,-}, u_{j_0^n-1/2}^{n,+})\right) &= C_{j_0^n+1/2}^{n,p} \Delta_+ \overline{u}_{j_0^n}^{n,p}\\ -D_{j_0^n-1/2}^n \Delta_- \overline{u}_{j_0^n}^n &- \left(h(u_{j_0^n+1/2}^{n,p,-}, u_{j_0^n-1/2}^{n,p,+}) - h(u_{j_0^n+1/2}^{n,-}, u_{j_0^n-1/2}^{n,+})\right),\end{aligned}$$

where $u_{j_0^n-1/2}^{n,p,+}$ is computed by the same predictor as $u_{j_0^n+1/2}^{n,p,-}$. In addition, we write

$$\begin{aligned}&-\left(h(u_{j_0^n+1/2}^{n,p,-}, u_{j_0^n-1/2}^{n,p,+}) - h(u_{j_0^n+1/2}^{n,-}, u_{j_0^n-1/2}^{n,+})\right)\\ &\quad = -\mathfrak{h}_{2,j_0^n-1/2}^{n,p} \cdot \left(u_{j_0^n-1/2}^{n,p,+} - u_{j_0^n-1/2}^{n,+}\right) - \mathfrak{h}_{1,j_0^n+1/2}^{n,p} \cdot \left(u_{j_0^n+1/2}^{n,p,-} - u_{j_0^n+1/2}^{n,-}\right),\end{aligned}$$

where

$$\begin{aligned}\mathfrak{h}_{1,j_0^n+1/2}^{n,p} &:= \tfrac{h(u_{j_0^n+1/2}^{n,p,-}, u_{j_0^n-1/2}^{n,+}) - h(u_{j_0^n+1/2}^{n,-}, u_{j_0^n-1/2}^{n,+})}{u_{j_0^n+1/2}^{n,p,-} - u_{j_0^n+1/2}^{n,-}} \geq 0,\\ -\mathfrak{h}_{2,j_0^n-1/2}^{n,p} &:= -\tfrac{h(u_{j_0^n+1/2}^{n,p,-}, u_{j_0^n-1/2}^{n,p,+}) - h(u_{j_0^n+1/2}^{n,p,-}, u_{j_0^n-1/2}^{n,+})}{u_{j_0^n-1/2}^{n,p,+} - u_{j_0^n-1/2}^{n,+}} \geq 0.\end{aligned} \tag{4.11}$$

Moreover, using the second order predictor (3.2), we deduce that

$$u_{j_0^n-1/2}^{n,p,+} - u_{j_0^n-1/2}^{n,+} = \frac{p}{M}\left(u_{j_0^n-1/2}^{n,(1),+} - u_{j_0^n-1/2}^{n,+}\right)$$

and

$$u_{j_0^n+1/2}^{n,p,-} - u_{j_0^n+1/2}^{n,-} = \frac{p}{M}\left(u_{j_0^n+1/2}^{n,(1),-} - u_{j_0^n+1/2}^{n,-}\right).$$

Therefore, we can rewrite the correction (4.10) as follows:

$$\begin{aligned}\widehat{\overline{u}}_{j_0^n}^{n,(1)} &= \overline{u}_{j_0^n}^n + \tfrac{\Delta t}{M\Delta x_{j_0^n}} \textstyle\sum_{p=0}^{M-1}\Big(C_{j_0^n+1/2}^{n,p} \Delta_+ \overline{u}_{j_0^n}^{n,p} - D_{j_0^n-1/2}^n \Delta_- \overline{u}_{j_0^n}^n\\ &\quad + \tfrac{p}{M}(-\mathfrak{h}_{2,j_0^n-1/2}^{n,p})\left(u_{j_0^n-1/2}^{n,(1),+} - u_{j_0^n-1/2}^{n,+}\right) - \tfrac{p}{M}\mathfrak{h}_{1,j_0^n+1/2}^{n,p}\left(u_{j_0^n+1/2}^{n,(1),+} - u_{j_0^n+1/2}^{n,+}\right)\Big).\end{aligned} \tag{4.12}$$

Similarly,

(4.13)
$$\overline{u}_{j_0^n}^{n+1} = \tfrac{1}{2}\overline{u}_{j_0^n}^{n} + \tfrac{1}{2}\Bigg[\widehat{\overline{u}}_{j_0^n}^{n,(1)} + \tfrac{\Delta t}{M\Delta x_{j_0^n}} \sum_{p=0}^{M-1} \Bigg(C_{j_0^n+1/2}^{n,p,(1)} \Delta_+ \overline{u}_{j_0^n}^{n,p,(1)} - D_{j_0^n-1/2}^{n,(1)} \Delta_- \overline{u}_{j_0^n}^{n,(1)} + \tfrac{p+1}{M}(-\mathfrak{h}_{2,j_0^n-1/2}^{n,p,(1)}) \left(u_{j_0^n-1/2}^{n,(1),+} - u_{j_0^n-1/2}^{n,+} \right) - \tfrac{p+1}{M} \mathfrak{h}_{1,j_0^n+1/2}^{n,p,(1)} \left(u_{j_0^n+1/2}^{n,(1),+} - u_{j_0^n+1/2}^{n,+} \right) \Bigg) \Bigg],$$

in which $\mathfrak{h}_{1,j_0^n+1/2}^{n,p,(1)}$ and $\mathfrak{h}_{2,j_0^n-1/2}^{n,p,(1)}$ are defined in a similar way to (4.11) but with the solutions of the first stage $u_{j_0^n\pm1/2}^{n,p,(1),\pm}$ and $u_{j_0^n\pm1/2}^{n,(1),\pm}$. The TVBM property of the second order LTS scheme is guaranteed by the following theorem.

Theorem 4.2 (TVBM)**.** *Assume that there exists some $\theta > 0$ such that*

(4.14)
$$\begin{aligned} &-\theta \le \frac{\Delta_+ \widetilde{\overline{u}}_j^{n,(i)}}{\Delta_+ \overline{u}_j^{n,(i)}} \le 1 \quad \forall\, j < j_0^n, \quad -\theta \le \frac{\Delta_+ \widetilde{\overline{u}}_j^{n,p,(i)}}{\Delta_+ \overline{u}_j^{n,p,(i)}} \le 1 \quad \forall\, j \ge j_0^n, \\ &-\theta \le -\frac{\Delta_+ \widetilde{u}_j^{n,(i)}}{\Delta_+ \overline{u}_j^{n,(i)}} \le 1 \quad \forall\, j < j_0^n, \quad -\theta \le -\frac{\Delta_+ \widetilde{u}_j^{n,p,(i)}}{\Delta_+ \overline{u}_j^{n,p,(i)}} \le 1 \quad \forall\, j \ge j_0^n, \end{aligned}$$

for $n = 0, \dots, N-1$, $p = 0, \dots, M-1$, and $i = 0, 1$. In addition, if a local CFL condition is satisfied:

$$\lambda_j^{n,p}(h_1 - h_2) \le \frac{1}{1+\theta}, \tag{4.15}$$

where h_1 and $-h_2$ are the Lipschitz coefficients of $h(\cdot,\cdot)$ with respect to the first and second arguments, respectively, and $\lambda_j^{n,p}$ is defined by

$$\lambda_j^{n,p} = \begin{cases} \frac{\Delta t}{\Delta x_j} & \text{if } j \le j_0^n + 1 \\ \frac{\Delta t}{M \Delta x_j} & \text{if } j > j_0^n + 1 \end{cases} \qquad \begin{array}{l} \text{for } n = 0, \dots, N-1 \\ \text{and } p = 0, \dots, M-1. \end{array}$$

Then the second order LTS scheme is TVBM.

Proof. Following the techniques in [39], we first introduce some important facts that will be used later in the proof. From the monotonicity of $h(\cdot,\cdot)$ and (4.14), we deduce that, in (4.6):

$$C_{j+1/2}^{n,(i)}, D_{j+1/2}^{n,(i)} \ge 0 \quad \forall j < j_0 \quad \text{and} \quad C_{j+1/2}^{n,p,(i)}, D_{j+1/2}^{n,p,(i)} \ge 0 \quad \forall j \ge j_0. \tag{4.16}$$

We may omit the superscripts for the ease of presentation. Given any two nonnegative numbers α, β and supposing $\lambda_j^{n,p} = \max(\alpha, \beta)$ satisfies (4.15), we have

$$\alpha C_{j+1/2} + \beta D_{j+1/2} \le 1,$$

and consequently,

$$\begin{aligned} &\left|\overline{u}_{j+1} - \overline{u}_j - \alpha D_{j+1/2}\Delta_- \overline{u}_{j+1} - \beta C_{j+1/2}\Delta_+ \overline{u}_j\right| = |\Delta_+ \overline{u}_j| \left|1 - \alpha D_{j+1/2} - \beta C_{j+1/2}\right| \\ &= |\overline{u}_{j+1} - \overline{u}_j| - \alpha D_{j+1/2} |\Delta_- \overline{u}_{j+1}| - \beta C_{j+1/2} |\Delta_+ \overline{u}_j|, \end{aligned}$$

in which the functions must be evaluated at the same time level. Then, together with (4.16), we deduce that

(4.17)
$$\begin{aligned} &\left|\overline{u}_{j+1} - \overline{u}_j + \alpha \left(C_{j+3/2}\Delta_+ \overline{u}_{j+1} - D_{j+1/2}\Delta_- \overline{u}_{j+1}\right) - \beta \left(C_{j+1/2}\Delta_+ \overline{u}_j - D_{j-1/2}\Delta_- \overline{u}_j\right)\right| \\ &\le \left|\overline{u}_{j+1} - \overline{u}_j - \alpha D_{j+1/2}\Delta_- \overline{u}_{j+1} - \beta C_{j+1/2}\Delta_+ \overline{u}_j\right| + \alpha C_{j+3/2} |\Delta_+ \overline{u}_{j+1}| + \beta D_{j-1/2} |\Delta_- \overline{u}_j| \\ &\le |\overline{u}_{j+1} - \overline{u}_j| + \alpha \left(C_{j+3/2} |\Delta_+ \overline{u}_{j+1}| - D_{j+1/2} |\Delta_- \overline{u}_{j+1}|\right) \\ &\qquad - \beta \left(C_{j+1/2} |\Delta_+ \overline{u}_j| - D_{j-1/2} |\Delta_- \overline{u}_j|\right). \end{aligned}$$

In the following, we compute the variation $|u_{j+1} - u_j|$ for all j. Particularly, we consider four cases: i) *If* $j < (j_0^n - 1)$. From (4.8), we find that

$$\begin{aligned}\overline{u}_{j+1}^{n,(1)} - \overline{u}_j^{n,(1)} = \quad & \left(\overline{u}_{j+1}^n - \overline{u}_j^n\right) + \tfrac{\Delta t}{\Delta x_{j+1}}\left(C_{j+3/2}^n \Delta_+ \overline{u}_{j+1}^n - D_{j+1/2}^n \Delta_- \overline{u}_{j+1}^n\right) \\ & - \tfrac{\Delta t}{\Delta x_j}\left(C_{j+1/2}^n \Delta_+ \overline{u}_j^n - D_{j-1/2}^n \Delta_- \overline{u}_j^n\right)\Big].\end{aligned} \tag{4.18}$$

Applying (4.17) with $\alpha = \frac{\Delta t}{\Delta x_{j+1}}$ and $\beta = \frac{\Delta t}{\Delta x_j}$, we deduce from (4.18) that

$$\left|\overline{u}_{j+1}^{n,(1)} - \overline{u}_j^{n,(1)}\right| \le \left|\overline{u}_{j+1}^n - \overline{u}_j^n\right| + \Delta_+\left[\tfrac{\Delta t}{\Delta x_j}\left(C_{j+1/2}^n \left|\Delta_+ \overline{u}_j^n\right| - D_{j-1/2}^n \left|\Delta_- \overline{u}_j^n\right|\right)\right].$$

From this we obtain

$$\begin{aligned}\left|\overline{u}_{j+1}^{n+1} - \overline{u}_j^{n+1}\right| &\le \frac{1}{2}\left|\overline{u}_{j+1}^n - \overline{u}_j^n\right| + \frac{1}{2}\left|\overline{u}_{j+1}^{n,(1)} - \overline{u}_j^{n,(1)}\right| \\ &\quad + \frac{1}{2}\Delta_+\left[\frac{\Delta t}{\Delta x_j}\left(C_{j+1/2}^{n,(1)}\left|\Delta_+ \overline{u}_j^{n,(1)}\right| - D_{j-1/2}^{n,(1)}\left|\Delta_- \overline{u}_j^{n,(1)}\right|\right)\right] \\ &\le \left|\overline{u}_{j+1}^n - \overline{u}_j^n\right| + \frac{1}{2}\sum_{\nu=0}^{1}\Delta_+\left[\frac{\Delta t}{\Delta x_j}\left(C_{j+1/2}^{n,(\nu)}\left|\Delta_+ \overline{u}_j^{n,(\nu)}\right| - D_{j-1/2}^{n,(\nu)}\left|\Delta_- \overline{u}_j^{n,(\nu)}\right|\right)\right]\end{aligned}$$

or, equivalently,

$$\begin{aligned}\left|\overline{u}_{j+1}^{n+1} - \overline{u}_j^{n+1}\right| \le \left|\overline{u}_{j+1}^n - \overline{u}_j^n\right| + \frac{1}{2}\sum_{p=0}^{M-1}\sum_{\nu=0}^{1}\Delta_+\Big[\frac{\Delta t}{M\Delta x_j}\big(C_{j+1/2}^{n,(\nu)}\left|\Delta_+ \overline{u}_j^{n,(\nu)}\right| \\ - D_{j-1/2}^{n,(\nu)}\left|\Delta_- \overline{u}_j^{n,(\nu)}\right|\big)\Big] \quad \forall\, j < j_0^n - 1.\end{aligned} \tag{4.19}$$

ii) *If* $j > j_0^n$. By the same argument applied to (4.9) with the fine time step size, we find that

$$\begin{aligned}&\left|\overline{u}_{j+1}^{n+1} - \overline{u}_j^{n+1}\right| \le \left|\overline{u}_{j+1}^{n,M-1} - \overline{u}_j^{n,M-1}\right| \\ &+\tfrac{1}{2}\textstyle\sum_{\nu=0}^{1}\Delta_+\left[\tfrac{\Delta t}{M\Delta x_j}\left(C_{j+1/2}^{n,M-1,(\nu)}\left|\Delta_+ \overline{u}_j^{n,M-1,(\nu)}\right| - D_{j-1/2}^{n,M-1,(\nu)}\left|\Delta_- \overline{u}_j^{n,M-1,(\nu)}\right|\right)\right].\end{aligned}$$

Repeating this argument inductively, we obtain a similar bound as (4.19):

$$\begin{aligned}\left|\overline{u}_{j+1}^{n+1} - \overline{u}_j^{n+1}\right| \quad &\le \left|\overline{u}_{j+1}^n - \overline{u}_j^n\right| + \tfrac{1}{2}\textstyle\sum_{p=0}^{M-1}\sum_{\nu=0}^{1}\Delta_+\Big[\tfrac{\Delta t}{M\Delta x_j}\big(C_{j+1/2}^{n,p,(\nu)}\left|\Delta_+ \overline{u}_j^{n,p,(\nu)}\right| \\ &\quad - D_{j-1/2}^{n,p,(\nu)}\left|\Delta_- \overline{u}_j^{n,p,(\nu)}\right|\big)\Big] \qquad \forall\, j > j_0^n.\end{aligned} \tag{4.20}$$

iii) *If* $j = j_0^n$ (*the interface cell*). We aim to show that (4.20) again holds for $j = j_0^n$, which is the main part of the proof. Using the formulation for the corrector

(4.12)-(4.13), as well as the time stepping scheme in the fine region, we obtain

$$\begin{aligned}
&\overline{u}^{n+1}_{j_0^n+1} - \overline{u}^{n+1}_{j_0^n} = \overline{u}^{n}_{j_0^n+1} - \overline{u}^{n}_{j_0^n} \\
&\quad + \frac{1}{2}\sum_{p=0}^{M-1}\sum_{\nu=0}^{1}\Bigg[\frac{\Delta t}{M\Delta x_{j_0^n+1}}\left(C^{n,p,(\nu)}_{j_0^n+3/2}\,\Delta_+\overline{u}^{n,p,(\nu)}_{j_0^n+1} - D^{n,p,(\nu)}_{j_0^n+1/2}\,\Delta_-\overline{u}^{n,p,(\nu)}_{j_0^n+1}\right) \\
&\quad - \frac{\Delta t}{M\Delta x_{j_0^n}}\left(C^{n,p,(\nu)}_{j_0^n+1/2}\Delta_+\overline{u}^{n,p,(\nu)}_{j_0^n} - D^{n,(\nu)}_{j_0^n-1/2}\Delta_-\overline{u}^{n,(\nu)}_{j_0^n}\right)\Bigg] \\
&\quad - \frac{1}{2}\sum_{p=0}^{M-1}\frac{\Delta t}{M\Delta x_{j_0^n}}\Bigg[\left(\frac{p}{M}(-\mathfrak{h}^{n,p}_{2,j_0^n-1/2}) + \frac{p+1}{M}(-\mathfrak{h}^{n,p,(1)}_{2,j_0^n-1/2})\right)\left(u^{n,(1),+}_{j_0^n-1/2} - u^{n,+}_{j_0^n-1/2}\right) \\
&\quad - \left(\frac{p}{M}\mathfrak{h}^{n,p}_{1,j_0^n+1/2} + \frac{p+1}{M}\mathfrak{h}^{n,p,(1)}_{1,j_0^n+1/2}\right)\left(u^{n,(1),-}_{j_0^n+1/2} - u^{n,-}_{j_0^n+1/2}\right)\Bigg)\Bigg].
\end{aligned}$$

Since $u^{n,(1),\pm}_{j_0^n-1/2} - u^{n,\pm}_{j_0^n-1/2} = O(\Delta t)$, and by the CFL condition, we can bound

(4.21)

$$\begin{aligned}
\left|\overline{u}^{n+1}_{j_0^n+1} - \overline{u}^{n+1}_{j_0^n}\right| \le \Bigg|&\overline{u}^{n}_{j_0^n+1} - \overline{u}^{n}_{j_0^n} + \frac{1}{2}\sum_{p=0}^{M-1}\sum_{\nu=0}^{1}\Bigg[\frac{\Delta t}{M\Delta x_{j_0^n+1}}\big(C^{n,p,(\nu)}_{j_0^n+3/2}\,\Delta_+\overline{u}^{n,p,(\nu)}_{j_0^n+1} \\
&- D^{n,p,(\nu)}_{j_0^n+1/2}\,\Delta_-\overline{u}^{n,p,(\nu)}_{j_0^n+1}\big) - \frac{\Delta t}{M\Delta x_{j_0^n}}\left(C^{n,p,(\nu)}_{j_0^n+1/2}\Delta_+\overline{u}^{n,p,(\nu)}_{j_0^n} - D^{n,(\nu)}_{j_0^n-1/2}\Delta_-\overline{u}^{n,(\nu)}_{j_0^n}\right)\Bigg]\Bigg| \\
&+ \frac{M}{1+\theta}O(\Delta t).
\end{aligned}$$

We have

(4.22)

$$\begin{aligned}
\overline{u}^{n}_{j_0^n+1} - \overline{u}^{n}_{j_0^n} = \frac{1}{2M}\sum_{p=0}^{M-1}\Bigg\{&\left[\left(\overline{u}^{n}_{j_0^n+1} - \overline{u}^{n}_{j_0^n}\right) - \left(\overline{u}^{n,p}_{j_0^n+1} - \overline{u}^{n,p}_{j_0^n}\right)\right] \\
&+\left[\left(\overline{u}^{n}_{j_0^n+1} - \overline{u}^{n}_{j_0^n}\right) - \left(\overline{u}^{n,p,(1)}_{j_0^n+1} - \overline{u}^{n,p,(1)}_{j_0^n}\right)\right] + \left(\overline{u}^{n,p}_{j_0^n+1} - \overline{u}^{n,p}_{j_0^n}\right) + \left(\overline{u}^{n,p,(1)}_{j_0^n+1} - \overline{u}^{n,p,(1)}_{j_0^n}\right)\Bigg\}.
\end{aligned}$$

Regarding the first two terms, let us write

(4.23)

$$\begin{aligned}
&\frac{1}{2}\left[\left(\overline{u}^{n}_{j_0^n+1} - \overline{u}^{n}_{j_0^n}\right) - \left(\overline{u}^{n,p}_{j_0^n+1} - \overline{u}^{n,p}_{j_0^n}\right)\right] + \frac{1}{2}\left[\left(\overline{u}^{n}_{j_0^n+1} - \overline{u}^{n}_{j_0^n}\right) - \left(\overline{u}^{n,p,(1)}_{j_0^n+1} - \overline{u}^{n,p,(1)}_{j_0^n}\right)\right] \\
&= \left(\overline{u}^{n}_{j_0^n+1} - \overline{u}^{n,p}_{j_0^n+1}\right) + \frac{1}{2}\left(\overline{u}^{n,p}_{j_0^n+1} - u^{n,p,(1)}_{j_0^n+1}\right) + \frac{1}{2}\left[\left(\overline{u}^{n,p}_{j_0^n} - \overline{u}^{n}_{j_0^n}\right) + \left(\overline{u}^{n,p,(1)}_{j_0^n} - \overline{u}^{n}_{j_0^n}\right)\right].
\end{aligned}$$

By definition of the second order predictor (3.2), the last term in (4.23) is given by

$$\frac{1}{2}\left[\left(\overline{u}^{n,p}_{j_0^n} - \overline{u}^{n}_{j_0^n}\right) + \left(\overline{u}^{n,p,(1)}_{j_0^n} - \overline{u}^{n}_{j_0^n}\right)\right] = \frac{2p+1}{2M}\left(\overline{u}^{n,(1)}_{j_0^n} - \overline{u}^{n}_{j_0^n}\right) = O(\Delta t). \tag{4.24}$$

On the other hand, the first and second terms in (4.23) can be computed by using the time stepping in the fine region (4.9):

$$\begin{aligned}
&\left(\overline{u}^{n}_{j_0^n+1} - \overline{u}^{n,p}_{j_0^n+1}\right) + \tfrac{1}{2}\left(\overline{u}^{n,p}_{j_0^n+1} - \overline{u}^{n,p,(1)}_{j_0^n+1}\right) \\
&\quad = -\tfrac{1}{2}\textstyle\sum_{q=0}^{p}\frac{\Delta t}{M\Delta x_{j_0^n+1}}\left(C^{n,q}_{j_0^n+3/2}\,\Delta_+\overline{u}^{n,q}_{j_0^n+1} - D^{n,q}_{j_0^n+1/2}\,\Delta_-\overline{u}^{n,q}_{j_0^n+1}\right) \\
&\qquad -\tfrac{1}{2}\textstyle\sum_{q=0}^{p-1}\frac{\Delta t}{M\Delta x_{j_0^n+1}}\left(C^{n,q,(1)}_{j_0^n+3/2}\,\Delta_+\overline{u}^{n,q,(1)}_{j_0^n+1} - D^{n,q,(1)}_{j_0^n+1/2}\,\Delta_-\overline{u}^{n,q,(1)}_{j_0^n+1}\right).
\end{aligned} \tag{4.25}$$

Summing (4.25) over $p = 0, \dots, M-1$ yields

(4.26)
$$\begin{aligned}
&\sum_{p=0}^{M-1} \left\{ \left(\overline{u}^n_{j_0^n+1} - \overline{u}^{n,p}_{j_0^n+1} \right) + \frac{1}{2} \left(\overline{u}^{n,p}_{j_0^n+1} - u^{n,p,(1)}_{j_0^n+1} \right) \right\} \\
&\quad = -\frac{1}{2} \sum_{p=0}^{M-1} \left(1 - \frac{p}{M} \right) \frac{\Delta t}{\Delta x_{j_0^n+1}} \left(C^{n,p}_{j_0^n+3/2} \, \Delta_+ \overline{u}^{n,p}_{j_0^n+1} - D^{n,p}_{j_0^n+1/2} \, \Delta_- \overline{u}^{n,p}_{j_0^n+1} \right) \\
&\qquad - \frac{1}{2} \sum_{p=0}^{M-1} \left(1 - \frac{p+1}{M} \right) \frac{\Delta t}{\Delta x_{j_0^n+1}} \left(C^{n,p,(1)}_{j_0^n+3/2} \, \Delta_+ \overline{u}^{n,p,(1)}_{j_0^n+1} - D^{n,p,(1)}_{j_0^n+1/2} \, \Delta_- \overline{u}^{n,p,(1)}_{j_0^n+1} \right).
\end{aligned}$$

Substituting (4.24) and (4.26) into (4.22) and then (4.21), and using (4.17) with $\alpha = \frac{p}{M} \left(\frac{\Delta t}{\Delta x_{j_0^n+1}} \right)$ or $\alpha = \frac{p+1}{M} \left(\frac{\Delta t}{\Delta x_{j_0^n+1}} \right)$ and $\beta = \frac{\Delta t}{\Delta x_{j_0^n}}$, we obtain:

(4.27)
$$\begin{aligned}
\left| \overline{u}^{n+1}_{j_0^n+1} - \overline{u}^{n+1}_{j_0^n} \right| \le{}& \tfrac{1}{2M} \textstyle\sum_{p=0}^{M-1} \left\{ \left| \overline{u}^{n,p}_{j_0^n+1} - \overline{u}^{n,p}_{j_0^n} \right| + \left| \overline{u}^{n,p,(1)}_{j_0^n+1} - \overline{u}^{n,p,(1)}_{j_0^n} \right| \right. \\
&+ \tfrac{p}{M} \tfrac{\Delta t}{\Delta x_{j_0^n+1}} \left(C^{n,p}_{j_0^n+3/2} \left| \Delta_+ \overline{u}^{n,p}_{j_0^n+1} \right| - D^{n,p}_{j_0^n+1/2} \left| \Delta_- \overline{u}^{n,p}_{j_0^n+1} \right| \right) \\
&+ \tfrac{p+1}{M} \tfrac{\Delta t}{\Delta x_{j_0^n+1}} \left(C^{n,p,(1)}_{j_0^n+3/2} \left| \Delta_+ \overline{u}^{n,p,(1)}_{j_0^n+1} \right| - D^{n,p,(1)}_{j_0^n+1/2} \left| \Delta_- \overline{u}^{n,p,(1)}_{j_0^n+1} \right| \right) \\
&\left. - \tfrac{\Delta t}{\Delta x_{j_0^n}} \textstyle\sum_{\nu=0}^{1} \left(C^{n,p,(\nu)}_{j_0^n+1/2} \left| \Delta_+ \overline{u}^{n,p,(\nu)}_{j_0^n} \right| - D^{n,(\nu)}_{j_0^n-1/2} \left| \Delta_- \overline{u}^{n,(\nu)}_{j_0^n} \right| \right) \right\} + O(\Delta t).
\end{aligned}$$

Furthermore, by the definition of the second order predictor, we have $\overline{u}^{n,p,(1)}_{j_0^n} = \overline{u}^{n,p}_{j_0^n} + \frac{1}{M} \left(\overline{u}^{n,(1)}_{j_0^n} - \overline{u}^{n}_{j_0^n} \right) = \overline{u}^{n,p}_{j_0^n} + O(\Delta t)$. This together with using (4.17) for $\alpha = \frac{\Delta t}{M \Delta x_{j_0^n+1}}$ and $\beta = 0$, we have that

(4.28)
$$\begin{aligned}
\left| \overline{u}^{n,p,(1)}_{j_0^n+1} - \overline{u}^{n,p,(1)}_{j_0^n} \right| &= \left| \overline{u}^{n,p}_{j_0^n+1} - \overline{u}^{n,p}_{j_0^n} \right. \\
&\quad \left. + \tfrac{\Delta t}{M \Delta x_{j_0^n+1}} \left(C^{n,p}_{j_0^n+3/2} \, \Delta_+ \overline{u}^{n,p}_{j_0^n+1} - D^{n,p}_{j_0^n+1/2} \, \Delta_- \overline{u}^{n,p}_{j_0^n+1} \right) \right| + O(\Delta t) \\
&\le \left| \overline{u}^{n,p}_{j_0^n+1} - \overline{u}^{n,p}_{j_0^n} \right| + \tfrac{\Delta t}{M \Delta x_{j_0^n+1}} \left(C^{n,p}_{j_0^n+3/2} \left| \Delta_+ \overline{u}^{n,p}_{j_0^n+1} \right| - D^{n,p}_{j_0^n+1/2} \left| \Delta_- \overline{u}^{n,p}_{j_0^n+1} \right| \right) + O(\Delta t).
\end{aligned}$$

Plugging this into (4.27) yields:

(4.29)
$$\begin{aligned}
\left| \overline{u}^{n+1}_{j_0^n+1} - \overline{u}^{n+1}_{j_0^n} \right| \le{}& \tfrac{1}{M} \textstyle\sum_{p=0}^{M-1} \left\{ \left| \overline{u}^{n,p}_{j_0^n+1} - \overline{u}^{n,p}_{j_0^n} \right| \right. \\
&+ \tfrac{p+1}{2M} \tfrac{\Delta t}{\Delta x_{j_0^n+1}} \left(C^{n,p}_{j_0^n+3/2} \left| \Delta_+ \overline{u}^{n,p}_{j_0^n+1} \right| - D^{n,p}_{j_0^n+1/2} \left| \Delta_- \overline{u}^{n,p}_{j_0^n+1} \right| \right) \\
&+ \tfrac{p+1}{2M} \tfrac{\Delta t}{\Delta x_{j_0^n+1}} \left(C^{n,p,(1)}_{j_0^n+3/2} \left| \Delta_+ \overline{u}^{n,p,(1)}_{j_0^n+1} \right| - D^{n,p,(1)}_{j_0^n+1/2} \left| \Delta_- \overline{u}^{n,p,(1)}_{j_0^n+1} \right| \right) \\
&\left. - \tfrac{1}{2} \textstyle\sum_{\nu=0}^{1} \tfrac{\Delta t}{\Delta x_{j_0^n}} \left(C^{n,p,(\nu)}_{j_0^n+1/2} \left| \Delta_+ \overline{u}^{n,p,(\nu)}_{j_0^n} \right| - D^{n,(\nu)}_{j_0^n-1/2} \left| \Delta_- \overline{u}^{n,(\nu)}_{j_0^n} \right| \right) \right\} + O(\Delta t).
\end{aligned}$$

On the other hand, by the SSP-RK$(2,2)$ time stepping in the fine cell (j_0^n+1) and using (4.28), we deduce that

$$\begin{aligned}\left|\overline{u}^{n,p}_{j_0^n+1}-\overline{u}^{n,p}_{j_0^n}\right| &= \left|\tfrac{1}{2}\left(\overline{u}^{n,p-1}_{j_0^n+1}-\overline{u}^{n,p-1}_{j_0^n}\right)+\tfrac{1}{2}\left(\overline{u}^{n,p-1,(1)}_{j_0^n+1}-\overline{u}^{n,p-1,(1)}_{j_0^n}\right)\right.\\ &+\tfrac{1}{2}\tfrac{\Delta t}{M\Delta x_{j_0^n+1}}\left(C^{n,p-1,(1)}_{j_0^n+3/2}\,\Delta_+\overline{u}^{n,p-1,(1)}_{j_0^n+1}-D^{n,p-1,(1)}_{j_0^n+1/2}\,\Delta_-\overline{u}^{n,p-1,(1)}_{j_0^n+1}\right)\\ &\left.+\tfrac{1}{2}\left(\overline{u}^{n,p-1}_{j_0^n}+\overline{u}^{n,p-1,(1)}_{j_0^n}-2\overline{u}^{n,p}_{j_0^n}\right)\right| \le \left|\overline{u}^{n,p-1}_{j_0^n+1}-\overline{u}^{n,p-1}_{j_0^n}\right|\\ &+\tfrac{1}{2}\tfrac{\Delta t}{M\Delta x_{j_0^n+1}}\textstyle\sum_{\nu=0}^{1}\left(C^{n,p-1,(\nu)}_{j_0^n+3/2}\left|\Delta_+\overline{u}^{n,p-1,(\nu)}_{j_0^n+1}\right|-D^{n,p-1,(\nu)}_{j_0^n+1/2}\left|\Delta_-\overline{u}^{n,p-1,(\nu)}_{j_0^n+1}\right|\right)+O(\Delta t),\end{aligned}$$

in which we have used the definition of the second order predictor to obtain

$$\frac{1}{2}\left(\overline{u}^{n,p-1}_{j_0^n}+\overline{u}^{n,p-1,(1)}_{j_0^n}-2\overline{u}^{n,p}_{j_0^n}\right)=\frac{1}{2}\left(\overline{u}^{n,p-1}_{j_0^n}-\overline{u}^{n,p}_{j_0^n}\right)=\frac{1}{2M}\left(\overline{u}^{n}_{j_0^n}-\overline{u}^{n,(1)}_{j_0^n}\right)=O(\Delta t).$$

Repeating the above argument inductively, we arrive at

$$\begin{aligned}\left|\overline{u}^{n,p}_{j_0^n+1}-\overline{u}^{n,p}_{j_0^n}\right| &\le \left|\overline{u}^{n}_{j_0^n+1}-\overline{u}^{n}_{j_0^n}\right|\\ &+\tfrac{1}{2}\tfrac{\Delta t}{M\Delta x_{j_0^n+1}}\textstyle\sum_{q=0}^{p-1}\sum_{\nu=0}^{1}\left(C^{n,q,(\nu)}_{j_0^n+3/2}\left|\Delta_+\overline{u}^{n,q,(\nu)}_{j_0^n+1}\right|-D^{n,q,(\nu)}_{j_0^n+1/2}\left|\Delta_-\overline{u}^{n,q,(\nu)}_{j_0^n+1}\right|\right)+O(\Delta t).\end{aligned}$$

Consequently,

$$\begin{aligned}\textstyle\sum_{p=0}^{M-1}\left|\overline{u}^{n,p}_{j_0^n+1}-\overline{u}^{n,p}_{j_0^n}\right| &\le M\left|\overline{u}^{n}_{j_0^n+1}-\overline{u}^{n}_{j_0^n}\right|\\ &+\tfrac{1}{2}\tfrac{\Delta t}{M\Delta x_{j_0^n+1}}\textstyle\sum_{p=0}^{M-1}\sum_{q=0}^{p-1}\sum_{\nu=0}^{1}\left(C^{n,q,(\nu)}_{j_0^n+3/2}\left|\Delta_+\overline{u}^{n,q,(\nu)}_{j_0^n+1}\right|\right.\\ &\qquad\left.-D^{n,q,(\nu)}_{j_0^n+1/2}\left|\Delta_-\overline{u}^{n,q,(\nu)}_{j_0^n+1}\right|\right)+O(\Delta t)\\ &\le M\left|\overline{u}^{n}_{j_0^n+1}-\overline{u}^{n}_{j_0^n}\right|+\tfrac{1}{2}\tfrac{\Delta t}{M\Delta x_{j_0^n+1}}\textstyle\sum_{p=0}^{M-1}\sum_{\nu=0}^{1}\left(1-\frac{p+1}{M}\right)\left(C^{n,p,(\nu)}_{j_0^n+3/2}\left|\Delta_+\overline{u}^{n,p,(\nu)}_{j_0^n+1}\right|\right.\\ &\qquad\left.-D^{n,p,(\nu)}_{j_0^n+1/2}\left|\Delta_-\overline{u}^{n,p,(\nu)}_{j_0^n+1}\right|\right)+O(\Delta t),\end{aligned}$$

where the last inequality is obtained by reversing the order of summation. Plugging this into (4.29), we find that

(4.30)
$$\begin{aligned}\left|\overline{u}^{n+1}_{j_0^n+1}-\overline{u}^{n+1}_{j_0^n}\right| &\le \left|\overline{u}^{n}_{j_0^n+1}-\overline{u}^{n}_{j_0^n}\right|\\ &+\tfrac{1}{2}\textstyle\sum_{p=0}^{M-1}\sum_{\nu=0}^{1}\left[\tfrac{\Delta t}{M\Delta x_{j_0^n+1}}\left(C^{n,p,(\nu)}_{j_0^n+3/2}\left|\Delta_+\overline{u}^{n,p,(\nu)}_{j_0^n+1}\right|-D^{n,p,(\nu)}_{j_0^n+1/2}\left|\Delta_-\overline{u}^{n,p,(\nu)}_{j_0^n+1}\right|\right)\right.\\ &\left.+\tfrac{\Delta t}{M\Delta x_{j_0^n}}\left(C^{n,p,(\nu)}_{j_0^n+1/2}\left|\Delta_+\overline{u}^{n,p,(\nu)}_{j_0^n}\right|-D^{n,(\nu)}_{j_0^n-1/2}\left|\Delta_-\overline{u}^{n,(\nu)}_{j_0^n}\right|\right)\right]+O(\Delta t).\end{aligned}$$

iv) *If* $j=j_0^n-1$. It remains to investigate the case of the interface cell and its neighbor in Ω_c^n. As for (4.21), we have

$$\begin{aligned}\left|\overline{u}^{n+1}_{j_0^n}-\overline{u}^{n+1}_{j_0^n-1}\right| &\le \left|\overline{u}^{n}_{j_0^n}-\overline{u}^{n}_{j_0^n-1}+\frac{1}{2}\sum_{p=0}^{M-1}\sum_{\nu=0}^{1}\frac{\Delta t}{M\Delta x_{j_0^n}}\left(C^{n,p,(\nu)}_{j_0^n+1/2}\,\Delta_+\overline{u}^{n,p,(\nu)}_{j_0^n}\right.\right.\\ &\left.\left.-D^{n,(\nu)}_{j_0^n-1/2}\,\Delta_-\overline{u}^{n,(\nu)}_{j_0^n}\right)-\frac{\Delta t}{M\Delta x_{j_0^n-1}}\left(C^{n,(\nu)}_{j_0^n-1/2}\Delta_+\overline{u}^{n,(\nu)}_{j_0^n-1}-D^{n,(\nu)}_{j_0^n-3/2}\Delta_-\overline{u}^{n,(\nu)}_{j_0^n-1}\right)\right|\\ &+\frac{M}{1+\theta}O(\Delta t).\end{aligned}$$

By performing similar manipulations as for the case $j = j_0^n$, one arrives at

$$\begin{aligned}\left|\overline{u}_{j_0^n}^{n+1} - \overline{u}_{j_0^n-1}^{n+1}\right| &\le \left|\overline{u}_{j_0^n}^{n} - \overline{u}_{j_0^n-1}^{n}\right| \\ &+\tfrac{1}{2}\sum_{p=0}^{M-1}\sum_{\nu=0}^{1}\left[\frac{\Delta t}{M\Delta x_{j_0^n}}\left(C_{j_0^n+1/2}^{n,p,(\nu)}\left|\Delta_+\overline{u}_{j_0^n}^{n,p,(\nu)}\right| - D_{j_0^n-1/2}^{n,(\nu)}\left|\Delta_-\overline{u}_{j_0^n}^{n,(\nu)}\right|\right)\right. \\ &\left.+\frac{\Delta t}{M\Delta x_{j_0^n-1}}\left(C_{j_0^n-1/2}^{n,(\nu)}\left|\Delta_+\overline{u}_{j_0^n-1}^{n,(\nu)}\right| - D_{j_0^n-3/2}^{n,(\nu)}\left|\Delta_-\overline{u}_{j_0^n-1}^{n,(\nu)}\right|\right)\right] + O(\Delta t).\end{aligned} \tag{4.31}$$

Finally, we combine (4.19), (4.20), (4.30), and (4.31) and obtain:

$$TV(\overline{u}^{n+1}) \le TV(\overline{u}^n) + O(\Delta t) \quad \text{or} \quad TV(\overline{u}^n) \le TV(\overline{u}^0) + CT.$$

Hence, the second order LTS scheme is TVBM. □

The condition (4.14) is fulfilled if the solution is limited by the *minmod* function m defined in (2.7) (see [7]). The scheme remains TVB when the modified *minmod* function $\widetilde{m}$ is used, which is achieved by Theorem 2.2 in [45] (see also [7, Lemma 2.3]). Finally, the TVB property of the means $\overline{u}_j$ can be passed to the whole solution u_h in the same manner as the RK-DG method [7, Propositition 2.11]. We remark that it is assumed that the solution near the time-dependent LTS interface is sufficiently smooth so that the condition (4.14) is satisfied in the region of the LTS interface without limiting. In practice, local time stepping should be coupled with adaptive spatial meshing to achieve computational efficiency and accuracy when dealing with hyperbolic conservation laws.

5. Numerical experiments

We consider several standard test cases of one-dimensional scalar conservation laws (Subsection 5.1) and the system of conservation laws (Subsection 5.2). We aim to verify the accuracy, mass conservation, and stability of the LTS schemes as predicted theoretically and compare them with the global time stepping (GTS) schemes. As a first step towards studying the behavior of the proposed schemes, we use a fixed LTS interface (i.e., $j_0^n = j_0$ for all n) in all the tests, instead of a time varying LTS interface as discussed in Section 3, and leave the investigation on parallel performance of the methods with space-time adaptive multiresolution meshes in two and three dimensions to future work. Note that for the prediction step, if the solution is discontinuous at the fixed LTS interface, then it is necessary to limit the solution on the interface with the coarse time step size $\mathbf{u}_{j_0}^{n,(i)(\mathrm{mod})}$ for $i = 1, \ldots, s-1$ before calculating the predicted interface values (3.1) at intermediate time levels.

5.1. Scalar conservation laws.

We first consider two model problems that obey the scalar conservation laws: the linear advection equation and Burgers' equation. For problems with a smooth solution, we confirm the convergence order in time of our LTS algorithms. The effectiveness of LTS algorithms is demonstrated by comparing with the GTS schemes in terms of accuracy and CFL conditions.

Example 1 (Linear problem)**.** We solve the linear advection problem with a smooth initial condition

$$u_t + u_x = 0, \quad u(x,0) = \sin \pi x, \tag{5.1}$$

in $-1 \le x \le 1$ with periodic boundary conditions. The exact solution is given by $u(x,t) = \sin\pi(x-t)$. The spatial domain is divided into two subdomains,

$\Omega_1 = [-1, 0]$ and $\Omega_2 = [0, 1]$. The mesh size and time step size in Ω_i are, respectively, Δx_i and Δt_i, which are fine when $i = 1$ and are coarse when $i = 2$:

$$\Delta x_1 = \frac{\Delta x_{\text{coarse}}}{M},\ \Delta x_2 = \Delta x_{\text{coarse}},\ \ \Delta t_1 = \frac{\mathcal{C}}{2k+1}\Delta x_1 = \frac{\Delta t_{\text{coarse}}}{M},\ \Delta t_2 = \Delta t_{\text{coarse}},$$

for $M = 1, 2, 4, 8$, and for $k = 1, 2, 3$ corresponding to the second, third, and fourth order LTS methods. The L^1 relative errors at $T = 2$ of the three LTS algorithms are listed in Table 1. We observe that for all schemes, the errors decrease as M increases; and the LTS schemes (with $M = 2, 4, 8$) preserve the order of convergence as in the GTS case ($M = 1$), regardless of how large M is.

TABLE 1. [Linear advection with a smooth initial condition] L^1 relative errors at $T = 2$ for different M. The rates of convergence "CR" for fixed M are shown in square brackets.

Linear advection							
Δx_{coarse}	M	RK-DG2		RK-DG3		RK-DG4	
		Rel. L^1 error	[CR]	Rel. L^1 error	[CR]	Rel. L^1 error	[CR]
1/5	1	5.70e-02	–	1.39e-03	–	4.43e-05	–
	2	3.59e-02	–	8.09e-04	–	2.40e-05	–
	4	3.13e-02	–	7.65e-04	–	2.32e-05	–
	8	3.04e-02	–	7.62e-04	–	2.33e-05	–
1/10	1	1.36e-02	[2.07]	1.66e-04	[3.07]	2.73e-06	[4.02]
	2	8.71e-03	[2.04]	9.76e-05	[3.05]	1.46e-06	[4.04]
	4	7.60e-03	[2.04]	9.20e-05	[3.06]	1.39e-06	[4.06]
	8	3.91e-03	[2.04]	9.17e-05	[3.06]	1.41e-06	[4.05]
1/20	1	3.30e-03	[2.04]	2.04e-05	[3.03]	1.71e-07	[4.00]
	2	2.14e-03	[2.03]	1.20e-05	[3.02]	9.07e-08	[4.01]
	4	1.87e-03	[2.02]	1.14e-05	[3.01]	8.66e-08	[4.01]
	8	1.81e-03	[2.03]	1.13e-05	[3.02]	8.83e-08	[4.00]
1/40	1	8.11e-04	[2.03]	2.54e-06	[3.01]	1.07e-08	[4.00]
	2	5.31e-04	[2.01]	1.49e-06	[3.01]	5.67e-09	[4.00]
	4	4.64e-04	[2.01]	1.41e-06	[3.02]	5.42e-09	[4.00]
	8	4.50e-04	[2.01]	1.41e-06	[3.00]	5.53e-09	[4.00]

Now to show that the LTS algorithms are stable with a local CFL condition, we still consider the linear problem (5.1) but with a discontinuous initial condition

$$u(x, 0) = \begin{cases} 2, & x \le -1, \\ -1, & x > -1. \end{cases}$$

We are interested in the behavior of the approximate solution near the discontinuity $x = -0.5$ at $T = 0.5$ and $x = 0$ (the LTS interface) at $T = 1$. Again, the fine region is $\Omega_1 = [-1, 0]$ and the coarse one is $\Omega_2 = [0, 1]$. We use the second order RK-DG method and consider three schemes as follows:

(1) coarse GTS scheme with a coarse global time step size $\Delta t = \Delta t_{\text{coarse}}$;
(2) fine GTS scheme with a fine global time step size $\Delta t = {}^{\Delta t_{\text{coarse}}}/_{M}$;
(3) LTS scheme with spatial variable time step sizes $\Delta t_1 = {}^{\Delta t_{\text{coarse}}}/_{M}$ and $\Delta t_2 = \Delta t_{\text{coarse}}$.

Note that the spatial mesh is refined in Ω_1 by a factor of M. Table 2 shows the L^1 relative errors of the three schemes at $T = 0.5$ and $T = 1$, respectively. It is seen that the coarse GTS becomes unstable as the spatial mesh is refined due to the violation of the CFL condition, while the LTS scheme, with a valid local CFL condition, gives stable solution with the same accuracy as the fine GTS scheme.

TABLE 2. [Linear advection with a discontinuous initial condition] L^1 relative errors at $T = 0.5$ and $T = 1$ of the RK-DG2 global time stepping (GTS) and local time stepping (LTS) schemes.

Linear advection, RK-DG2							
Δx_{coarse}	M	At $T = 0.5$			At $T = 1$		
		coarse GTS	fine GTS	LTS	coarse GTS	fine GTS	LTS
1/40	1	2.58e-02			2.53e-02		
	2	3.46e-02	1.52e-02		3.46e-02	1.50e-02	
	4	–	9.00e-03		–	8.99e-03	
	8	–	5.40e-03		–	5.46e-03	

Example 2 (Burgers' equation)**.** Next, we test the proposed algorithms on the Burgers' equation with a smooth initial condition:

$$u_t + \left(\frac{u^2}{2}\right)_x = 0, \quad u(x,0) = \frac{1}{4} + \frac{1}{2}\sin \pi x, \tag{5.2}$$

in $-1 \le x \le 1$. The exact solution of the problem is given by [26]:

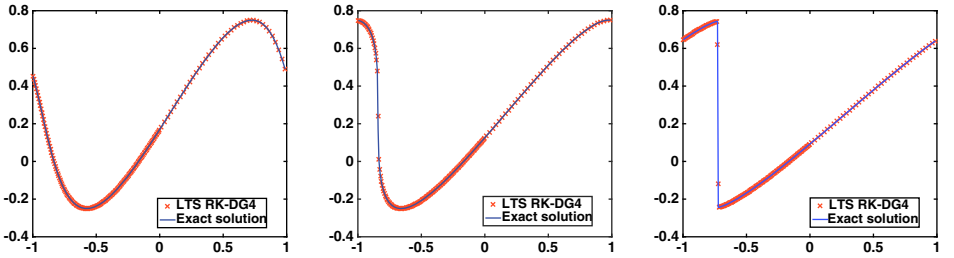
$$w(x,t) = \frac{1}{4} + \frac{1}{2}\, v(x - t, \frac{t}{2}), \tag{5.3}$$

in which $v(x,t)$ is the solution of the Burgers' equation with $v(x,0) = \sin \pi x$. We compute v by Newton iterations to solve the characteristic relation:

$$v = \sin(\pi x - vt), \quad 0 \le x < 1.$$

The solution v in $(-1, 0)$ is computed from v in $(0, 1)$ via: $v(-x,t) = -v(x,t)$. The solution of (5.2) is smooth up to $t = 2/\pi$; then it develops a moving shock. For details, see [26].

We divide the spatial domain into two zones and use the same discretization in space and in time as in Example 1. In Figure 2, we show the exact solution and the approximate solution by the fourth order LTS algorithm with $\Delta x_{\text{coarse}} = 1/40$ and $M = 4$. We see that the LTS scheme clearly captures the shock with local refinement in space and in time. In Figure 3, mass evolution as a function of time of different LTS schemes with $\Delta x_{\text{coarse}} = 1/40$ and $M = 4$ is displayed. The LTS schemes conserve the mass in the region of the LTS interface, and thus in the whole domain. The relative L^1 errors at $T = 0.3$ when the solution is still smooth are shown in Table 3. Again, the LTS schemes converge at the same order as the associated GTS schemes and the errors are improved as M increases. At $T = 1.1$, the errors of different LTS schemes in the smooth regions (0.1 away from the shock) have the same magnitude as the GTS approximation error as displayed in Table 4.



at $T = 0.3$ at $T = 2/\pi$ at $T = 1.1$

FIGURE 2. [Burgers' equation] Snapshots of the solution by the fourth order LTS scheme with 2 subdomains, $\Delta x_1 = \Delta x_2/4$ and $\Delta x_2 = 1/40$,

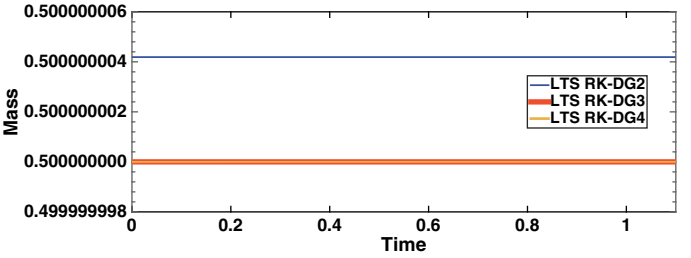


FIGURE 3. [Burgers' equation] Time evolution of mass for the second, third, and fourth order LTS schemes at $\Delta x_{\text{coarse}} = 1/40$ and $M = 4$. Note that the third and fourth order approximations coincide with each other.

Next we consider the case of multiple levels of refinement as explained in Remark 3.1. The spatial domain is divided into five equal intervals:

$$\Omega_1 = [-1, -0.6],\ \Omega_2 = [-0.6, -0.2], \ldots, \Omega_5 = [0.6, 1],$$

with different mesh sizes $\Delta x_i = \Delta x_{\text{coarse}}/2^{5-i}$ for $i = 1, \ldots, 5$. Different time step sizes are imposed that are determined by local CFL conditions. In particular, the local time step size Δt_1 is the smallest and $\Delta t_{i+1} = 2\Delta t_i$ for $i = 1, \ldots, 4$. To illustrate the performance of the proposed LTS algorithm for this multiple time stepping case, we only consider the second order RK-DG scheme, but the extension to higher order schemes is straightforward. The L^1 errors at $T = 0.3$ and $T = 1.1$ are shown in Table 5 with expected accuracy and convergence orders. The snapshots of solution at different times are depicted in Figure 4 with $\Delta x_{\text{coarse}} = 1/20$. The time evolution of mass plotted in Figure 5 for $\Delta x_{\text{coarse}} = 1/80$ confirms that the LTS scheme also conserves mass up to machine precision for the case of multiple levels of refinement.

5.2. **Euler equations of gas dynamics.** We next apply the proposed LTS algorithms to solve a system of one-dimensional conservation laws. For the spatial discretization, we employ the DG methods for systems of equations presented in [8] with the local projection limiting in the characteristic fields. The time stepping is

TABLE 3. [Burgers' equation] L^1 relative errors at $T = 0.3$ for different M. The rates of convergence "CR" for fixed M are shown in square brackets.

Δx_{coarse}	M	RK-DG2		RK-DG3		RK-DG4	
		Rel. L^1 error	[CR]	Rel. L^1 error	[CR]	Rel. L^1 error	[CR]
1/10	1	6.49e-03	–	3.27e-04	–	2.26e-05	–
	2	3.97e-03	–	1.10e-04	–	6.75e-06	–
	4	3.37e-03	–	8.97e-05	–	5.97e-06	–
	8	3.22e-03	–	8.73e-05	–	5.93e-06	–
1/20	1	1.62e-03	[2.00]	4.10e-05	[3.00]	1.31e-06	[4.11]
	2	9.63e-04	[2.04]	1.45e-05	[2.92]	4.19e-07	[4.01]
	4	8.11e-04	[2.06]	1.17e-05	[2.94]	3.60e-07	[4.05]
	8	7.74e-04	[2.06]	1.14e-05	[2.94]	3.57e-07	[4.05]
1/40	1	4.03e-04	[2.01]	5.01e-06	[3.03]	8.68e-08	[3.92]
	2	2.38e-04	[2.02]	1.81e-06	[3.00]	2.63e-08	[3.99]
	4	1.99e-04	[2.03]	1.43e-06	[3.03]	2.30e-08	[3.97]
	8	1.89e-04	[2.03]	1.40e-06	[3.03]	2.28e-08	[3.97]
1/80	1	1.00e-04	[2.01]	6.18e-07	[3.02]	5.44e-09	[4.00]
	2	5.95e-05	[2.00]	2.25e-07	[3.01]	1.75e-09	[3.91]
	4	4.95e-05	[2.01]	1.78e-07	[3.01]	1.56e-09	[3.88]
	8	4.71e-05	[2.01]	1.74e-07	[3.01]	1.55e-09	[3.88]

TABLE 4. [Burgers' equation] L^1 relative errors in smooth regions at $T = 1.1$ of different local time stepping schemes.

At $T = 1.1$, errors in smooth regions $\|x - \text{shock}\| >= 0.1$				
Δx_{coarse}	M	2nd order LTS	3rd order LTS	4th order LTS
1/40	1	6.16e-05	4.25e-07	4.31e-09
	2	4.67e-05	1.73e-07	1.71e-09
	4	4.37e-05	1.53e-07	1.59e-09
	8	4.35e-05	1.52e-07	1.60e-09

still SSP-RK and thus it is straightforward to apply the proposed LTS algorithms for such a system. We consider the Euler equations of gas dynamics for a polytropic gas:

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = \mathbf{0}, \quad \mathbf{u} = (\rho, m, E)^T, \quad \mathbf{f}(\mathbf{u}) = q\mathbf{u} + (0, P, qP)^T, \tag{5.4}$$

with $P = (\gamma - 1)\left(E - 1/2\rho q^2\right)$. Here ρ, q, P, and E are the density, velocity, pressure, and total energy, respectively; $m = \rho\, q$ is the momentum and γ is the ratio of specific heats. In the following computation, we use $\gamma = 1.4$ and present numerical results of applying the second order LTS algorithm to solve Riemann problems of Euler equations and the problem of interaction of blast waves. Note that for these test cases, there is no advantage of using higher order schemes as investigated in [8].

TABLE 5. [Burgers' equation] L^1 relative errors at $T = 0.3$ and $T = 1.1$ for the case of five levels of refinement with $\Delta x_i = \Delta x_{\text{coarse}}/2^{5-i}$ for $i = 1, \ldots, 5$.

Δx_{coarse}	At $T = 0.3$		At T=1.1			
	Rel. L^1 error	[CR]	Rel. L^1 error in smooth regions	[CR]	Rel. L^1 error globally	[CR]
1/20	7.80e-04	–	8.85e-05	–	1.73e-03	–
1/40	1.98e-04	[1.98]	2.51e-05	[1.82]	8.54e-04	[1.02]
1/80	4.77e-05	[2.05]	6.32e-06	[1.99]	4.19e-04	[1.03]
1/160	1.13e-05	[2.08]	1.59e-06	[1.99]	2.06e-04	[1.02]

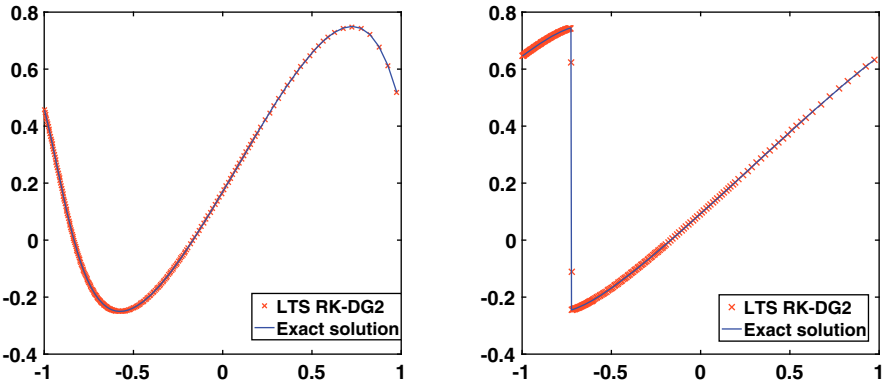


FIGURE 4. [Burgers' equation] Snapshots of the solution at $T = 0.3$ and $T = 1.1$ for the case of five levels of refinement with $\Delta x_i = \Delta x_{\text{coarse}}/2^{5-i}$ for $i = 1, \ldots, 5$. Here $\Delta x_{\text{coarse}} = 1/20$.

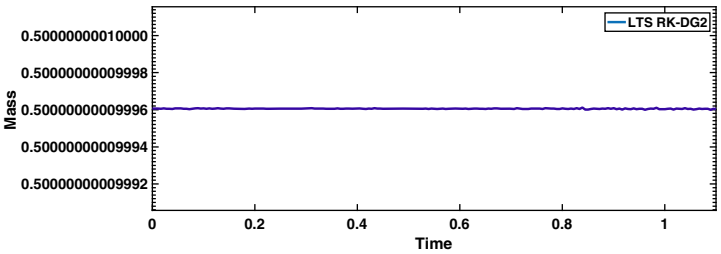


FIGURE 5. [Burgers' equation] Time evolution of mass obtained by the second order LTS on meshes with five resolutions $\Delta x_i = \Delta x_{\text{coarse}}/2^{5-i}$ for $i = 1, \ldots, 5$. Here $\Delta x_{\text{coarse}} = 1/80$.

Example 3 (Shock tube problem)**.** Consider the Riemann problem

$$\mathbf{u}(x,0) = \begin{cases} \mathbf{u}_L, & x < 0, \\ \mathbf{u}_R, & x > 0, \end{cases} \tag{5.5}$$

with two sets of initial conditions:

a) The Sod problem [50]: $(\rho_L, q_L, P_L) = (1, 0, 1)$ and $(\rho_R, q_R, P_R) = (0.125, 0, 0.10)$.
b) The Lax problem [33]: $(\rho_L, q_L, P_L) = (0.445, 0.698, 3.528)$ and $(\rho_R, q_R, P_R) = (0.5, 0, 0.571)$.

The Sod problem has become a standard test problem of Euler equations with a monotone decreasing density profile. For this problem, we consider two settings of the decomposition into fine and coarse regions:

(1) two subdomains with the coarse region $[-4.9, -0.5)$ and fine region $[-0.5, 5.1)$;
(2) three subdomains with the coarse regions $[-4.9, -2.9)$ and $[4, 5.1]$, and the fine region $[-2.9, 4)$.

The exact solution and approximation solution by the second order LTS algorithm at $T = 2.0$ with $\Delta x_{\text{coarse}} = 1/5$ and $M = 4$ are shown in Figure 6 for the case of two subdomains and in Figure 7 for the case of three subdomains. The L^1 relative errors are displayed in Table 6 in which we observe that for both settings, the errors decrease as M increases, especially for the three subdomain case, and the order of convergence is first order as the solution is discontinuous. The three subdomain setting, as expected, has a better performance since the fine region includes the contact discontinuities and the corners of rarefaction waves.

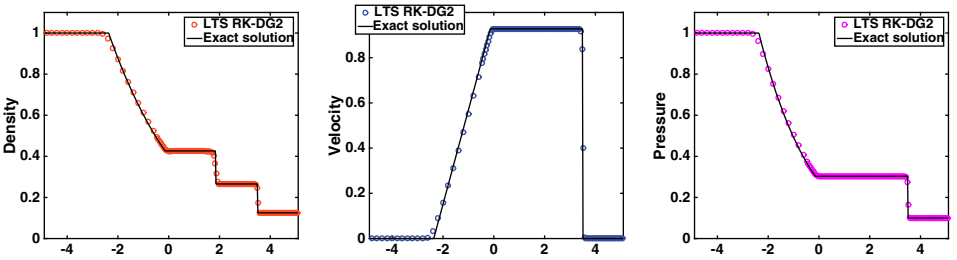


FIGURE 6. [Sod shock tube problem] Snapshots of the density, velocity, and pressure at $T = 2.0$ by the second order LTS scheme with 2 subdomains, $\Delta x_{\text{coarse}} = 1/5$ and $M = 4$.

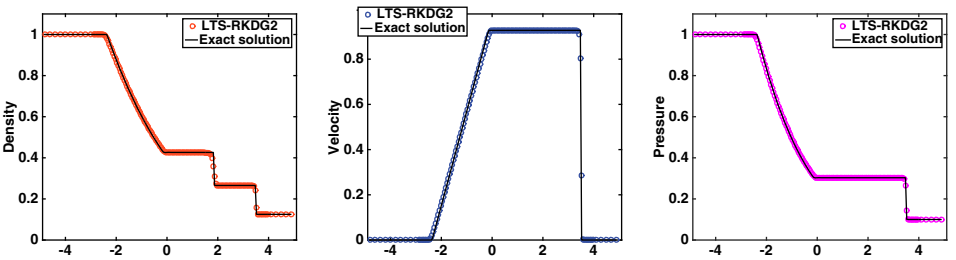


FIGURE 7. [Sod shock tube problem] Snapshots of the density, velocity, and pressure at $T = 2.0$ by the second order LTS scheme with 3 subdomains, $\Delta x_{\text{coarse}} = 1/5$ and $M = 4$.

TABLE 6. [Sod shock tube problem] L^1 relative errors at $T = 2.0$ of the second order LTS algorithm for the Sod shock tube problem.

Δx_{coarse}	M	Density		Velocity		Pressure	
		2 domains	3 domains	2 domains	3 domains	2 domains	3 domains
1/5	1	1.79e-02		3.88e-02		1.76e-02	
	2	8.96e-03	8.28e-03	2.03e-02	1.75e-02	8.46e-03	7.22e-03
	4	6.35e-03	4.25e-03	1.39e-02	8.82e-03	6.82e-03	3.62e-03
	8	6.06e-03	2.22e-03	1.27e-02	4.46e-03	7.34e-03	1.81e-03
1/10	1	8.28e-03		1.78e-02		7.24e-03	
	2	4.41e-03	4.23e-03	9.79e-03	8.83e-03	3.89e-03	3.61e-03
	4	2.88e-03	2.21e-03	6.11e-03	4.46e-03	2.76e-03	1.80e-03
	8	2.50e-03	1.16e-03	5.19e-03	2.30e-03	2.80e-03	9.03e-04

For the Lax problem, the density profile has a "built-up" intermediate state, thus we divide the domain into two subdomains with the coarse region $[-4.9, 0)$ and the fine region $[0, 5.1)$. The second order LTS approximate solution at $T = 1.3$ with $\Delta x_{\text{coarse}} = 1/5$ and $M = 4$ is shown in Figure 8 together with the exact solution. We observe that the LTS scheme captures very well the "built-up" density profile with local refinement in space and in time. The L^1 relative errors are presented in Table 7 that match our expectation.

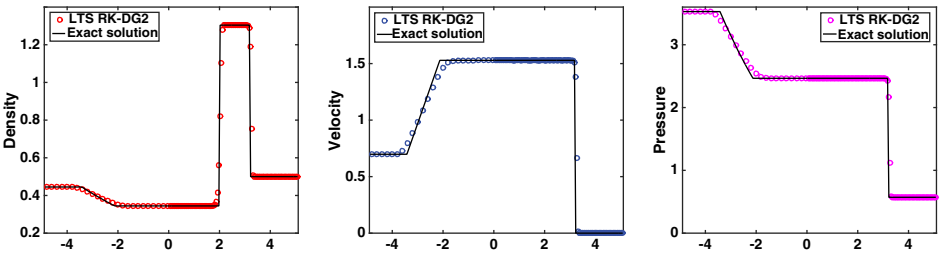


FIGURE 8. [Lax shock tube problem] Snapshots of the density, velocity, and pressure at $T = 1.3$ by the second order LTS scheme with two subdomains, $\Delta x_{\text{coarse}} = 1/5$ and $M = 4$.

Example 4 (Interaction of blast waves)**.** We finally consider the problem of two interacting blast waves:

$$\mathbf{u}(x,0) = \begin{cases} \mathbf{u}_L, & 0 \le x < 0.1, \\ \mathbf{u}_M, & 0.1 \le x < 0.9, \\ \mathbf{u}_R, & 0.9 \le x < 1, \end{cases} \tag{5.6}$$

with $(\rho_L, q_L, P_L) = (1, 0, 10^3)$, $(\rho_M, q_M, P_M) = (1, 0, 10^{-2})$, and $(\rho_R, q_R, P_R) = (1, 0, 100)$. Reflection boundary conditions are applied at $x = 0$ and $x = 1$. For details, see [26, 57].

We divide $\Omega = [0, 1]$ into 3 subdomains $\Omega_1 = [0, 0.2)$, $\Omega_2 = [0.2, 0.9)$, and $\Omega_3 = [0.9, 1]$. The mesh and time step sizes in Ω_2 are M times smaller than those in Ω_1 and Ω_3. The solutions at $T = 0.038$ with the second order global time

TABLE 7. [Lax shock tube problem] L^1 relative errors at $T = 1.3$ of the second order LTS algorithm.

Δx_{coarse}	M	Density	Velocity	Pressure
1/5	1	4.74e-02	2.15e-02	1.44e-02
	2	2.94e-02	1.69e-02	1.07e-02
	4	2.11e-02	1.73e-02	9.12e-03
	8	1.18e-02	1.37e-02	7.63e-03
1/10	1	2.52e-02	1.20e-02	6.55e-03
	2	1.51e-02	9.07e-03	4.74e-03
	4	8.88e-03	7.13e-03	3.91e-03
	8	5.70e-03	7.41e-03	4.19e-03

stepping ($M = 1$) and local time stepping ($M = 2$) are shown in Figure 9. We see that the LTS scheme with a local refinement in space and time gives a much better resolution, especially for the density profile.

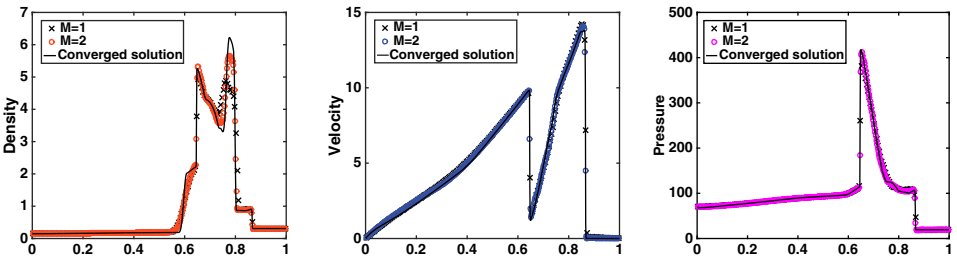


FIGURE 9. [Blast waves' interaction problem] Snapshots of the density, velocity, and pressure at $T = 0.038$ by the second order GTS ($M = 1$) and LTS ($M = 2$) schemes with $\Delta x_{\text{coarse}} = 1/200$.

6. CONCLUSION

In this work, high order explicit local time stepping algorithms have been proposed and analyzed for hyperbolic conservation laws. The approaches are of predictor-corrector-type, and algorithms of up to fourth order accuracy are constructed in a general setting of Runge-Kutta discontinuous Galerkin methods with the modified *minmod* limiter. With our LTS schemes, different time-step sizes can be used based on a local CFL condition instead of the more restrictive global CFL condition. Thus, they outperform the global time stepping for simulations on multiresolution meshes or of multiple scales. In addition, we rigorously prove the conservation property and nonlinear stability of these schemes. Numerical results confirm their accuracy and efficiency. Future work includes the coupling of adaptive multiresolution meshes with our local time stepping to carry out simulations in parallel and further investigations on the numerical performance of the proposed approaches for large scale simulations on modern supercomputer systems.

Appendices

A. SSP-RK time stepping schemes

We first present the SSP-RK(2,2) and SSP-RK(3,3) schemes for solution of the system (2.4), which are *optimal* in the sense that the number of stages equals the order of accuracy and the coefficients $\beta_{i,\nu}$ are nonnegative [20]. In both cases, the schemes possess the SSP coefficient $\mathcal{C} = 1$.

Second order SSP-RK(2,2): $\alpha_{10} = \beta_{10} = 1$, $\alpha_{20} = \alpha_{21} = \beta_{21} = 1/2$ and $\beta_{20} = 0$, which is equivalent to the Heun's method:

$$\begin{aligned} \mathbf{U}_h^{n,(1)} &= \mathbf{U}_h^n + \Delta t \mathbf{L}_h(\mathbf{U}_h^n), \\ \mathbf{U}_h^{n+1} &= 1/2\, \mathbf{U}_h^n + 1/2 \left(\mathbf{U}_h^{n,(1)} + \Delta t \mathbf{L}_h(\mathbf{U}_h^{n,(1)}) \right). \end{aligned} \tag{A.1}$$

Third order SSP-RK(3,3): $\alpha_{10} = \beta_{10} = 1$, $\alpha_{20} = 3/4$, $\alpha_{21} = \beta_{21} = 1/4$, $\alpha_{30} = 1/3$, $\alpha_{32} = \beta_{32} = 2/3$, and $\beta_{20} = \beta_{30} = \alpha_{31} = \beta_{31} = 0$, or explicitly:

$$\begin{aligned} \mathbf{U}_h^{n,(1)} &= \mathbf{U}_h^n + \Delta t \mathbf{L}_h(\mathbf{U}_h^n), \\ \mathbf{U}_h^{n,(2)} &= 3/4\, \mathbf{U}_h^n + 1/4 \left(\mathbf{U}_h^{n,(1)} + \Delta t \mathbf{L}_h(\mathbf{U}_h^{n,(1)}) \right), \\ \mathbf{U}_h^{n+1} &= 1/3\, \mathbf{U}_h^n + 2/3 \left(\mathbf{U}_h^{n,(2)} + \Delta t \mathbf{L}_h(\mathbf{U}_h^{n,(2)}) \right). \end{aligned} \tag{A.2}$$

For higher order schemes $r \geq 4$, we cannot avoid negative $\beta_{i\nu}$ without using additional stages. We shall use the SSP-RK(5,4) proposed in [32] with the SSP coefficient $\mathcal{C} = 1.652$, and the coefficients of the scheme are listed in Table 8.

Table 8. [32] Coefficients of the SSP-RK (5,4) scheme with $\mathcal{C} = 1.652$.

$\alpha_{i\nu}$				
1	0	0	0	0
0.261216512493821	0.738783487506179	0	0	0
0.623613752757655	0	0.376386247242345	0	0
0.444745181201454	0.120932584902288	0	0.434322233896258	0
0.213357715199957	0.209928473023448	0.063353148180384	0	0.513360663596212
$\beta_{i\nu}$				
0.605491839566400	0	0	0	0
0	0.447327372891397	0	0	0
0.000000844149769	0	0.227898801230261	0	0
0.002856233144485	0.073223693296006	0	0.262978568366434	0
0.002362549760441	0.127109977308333	0.038359814234063	0	0.310835692561898

B. Derivation of the predictors

We derive the predictors up to the fourth order accuracy ($k = 1, 2, 3$) in the DG-RK setting. To simplify the notation, in the following, we consider an arbitrary equation in the system (2.4) and write it in the form:

$$\partial_t w_j = L_j(\mathbf{w}), \tag{B.1}$$

where $\mathbf{w} = (w_j(t))_{\forall j}$ (w_j represents $u_j^{(l)}$ and we have dropped the superscript (l) for simplicity) and L_j is a multivariable, real-valued function.

Given the time partition with coarse and fine time steps as defined in Section 3 and assuming that the solution $\mathbf{w}^n$ at t^n is known, we shall construct the approximation of $\mathbf{w}$ at the interface $x_{j+1/2}$ at the intermediate time levels $t^{n,p}$ for

$p = 1, \ldots, M-1$. Performing Taylor expansion of w_j at t^n yields:

$$w_j(t) = w_j(t^n) + (t-t^n)\frac{dw_j}{dt}(t^n) + \ldots + \frac{1}{k!}(t-t^n)^k \frac{d^{(k)}w_j}{dt}(t^n) + O\left((\Delta t)^{k+1}\right). \tag{B.2}$$

The time derivatives of w_j up to order k are approximated by the SSP-RK solution of the first $(s-1)$ stages with the coarse time step size, $w_j^{n,(i)}$ for $i = 1, \ldots, s-1$. These approximations are detailed in the following for the second, third, and fourth order schemes, respectively.

B.1. **Predictor for the SSP-RK(2,2) scheme.** We obtain the approximation for $w_j^{n,p} = w_j^{n,p,(0)}$ by truncating (B.2) to the second term:

$$w_j^{n,p,(0)} = w_j^{n,(0)} + \frac{p\Delta t}{M}\partial_t w_j^{n,(0)} + O(\Delta t_p^2), \quad p = 0, 1, \ldots, M-1, \tag{B.3}$$

where $\Delta t_p = p^{\Delta t}/_M$. To compute the first time derivative, we first notice that $\partial_t w_j^{n,(0)} = L_j(\mathbf{w}^{n,(0)})$ according to (B.1). Thus, by using the solution at stage 1 of SSP-RK(2,2) with the coarse time step size

$$w_j^{n,(1)} = w_j^{n,(0)} + \Delta t L_j(\mathbf{w}^{n,(0)}),$$

we deduce that

$$\partial_t w_j^{n,(0)} = L_j(\mathbf{w}^{n,(0)}) = \frac{w_j^{n,(1)} - w_j^{n,(0)}}{\Delta t} + O\left(\Delta t\right). \tag{B.4}$$

Substituting this into (B.3) yields:

$$w_j^{n,p,(0)} = w_j^{n,(0)} + \frac{p}{M}\left(w_j^{n,(1)} - w_j^{n,(0)}\right) + O(\Delta t^2). \tag{B.5}$$

We also need to predict $w_j^{n,p,(1)}$, the solution at stage 1 at intermediate time levels, for $p = 0, \ldots, M-1$. By definition of the SSP-RK(2,2) scheme, we have:

$$w_j^{n,p,(1)} = w_j^{n,p,(0)} + \frac{\Delta t}{M} L_j(\mathbf{w}^{n,p,(0)}).$$

As $L_j(\mathbf{w}^{n,p,(0)}) = L_j(\mathbf{w}^{n,(0)}) + O(\Delta t)$, we deduce from (B.6) that

$$w_j^{n,p,(1)} = w_j^{n,p,(0)} + \frac{\Delta t}{M} L_j(\mathbf{w}^{n,(0)}) + O(\Delta t^2),$$

or, equivalently, via (B.5)

$$w_j^{n,p,(1)} = w_j^{n,(0)} + \frac{p+1}{M}\left(w_j^{n,(1)} - w_j^{n,(0)}\right) + O(\Delta t^2). \tag{B.6}$$

It is clear from (B.5) and (B.6) that the proposed predictor gives second order accurate approximations of the solutions at intermediate time levels $t^{n,p}$, for $p = 0, \ldots, M-1$.

B.2. **Predictor for the SSP-RK(3,3) scheme.** As in the second order case, we approximate $w_j^{n,p,(0)}$ by truncating (B.2), but now to the third term:

$$w_j^{n,p,(0)} = w_j^{n,(0)} + \frac{p\Delta t}{M}\partial_t w_j^{n,(0)} + \frac{1}{2}\left(\frac{p\Delta t}{M}\right)^2 \partial_{tt} w_j^{n,(0)} + O(\Delta t_p^3), \quad p = 0, 1, \ldots, M-1. \tag{B.7}$$

The time derivatives are computed from the solutions at stage 1 and stage 2 of SSP-RK(3,3) with the coarse time step size:

$$w_j^{n,(1)} = w_j^{n,(0)} + \Delta t L_j(\mathbf{w}^{n,(0)}), \tag{B.8}$$

$$w_j^{n,(2)} = \frac{3}{4} w_j^{n,(0)} + \frac{1}{4} w_j^{n,(1)} + \frac{1}{4} \Delta t L_j(\mathbf{w}^{n,(1)}). \tag{B.9}$$

The first time derivative can be obtained as in (B.4), for the second time derivative, by the chain rule we deduce from (B.1) that

$$\partial_{tt} w_j = \nabla L_j(\mathbf{w}) \cdot \partial_t \mathbf{w},$$

and we will compute the right-hand side by using (B.9). In particular, by performing first order Taylor expansion, we obtain:

$$L_j(\mathbf{w}^{n,(1)}) = L_j(\mathbf{w}^{n,(0)}) + \nabla L_j(\mathbf{w}^{n,(0)}) \cdot (\mathbf{w}^{n,(1)} - \mathbf{w}^{n,(0)}) + O(\Delta t^2).$$

Substituting this into (B.9) yields

$$w_j^{n,(2)} = \frac{3}{4} w_j^{n,(0)} + \frac{1}{4} w_j^{n,(1)} + \frac{1}{4} \Delta t L_j(\mathbf{w}^{n,(0)}) + \frac{1}{4} \Delta t^2 \nabla L_j(\mathbf{w}^{n,(0)}) \cdot \partial_t(\mathbf{w}^{n,(0)}) + O(\Delta t^3),$$

from which we deduce that

$$\begin{aligned} \Delta t^2 \nabla L_j(\mathbf{w}^{n,(0)}) \cdot \partial_t(\mathbf{w}^{n,(0)}) &= 4 w_j^{n,(2)} - 3 w_j^{n,(0)} - w_j^{n,(1)} - \Delta t L_j(\mathbf{w}^{n,(0)}) + O(\Delta t^3) \\ &= 4 w_j^{n,(2)} - 2 w_j^{n,(0)} - 2 w_j^{n,(1)} + O(\Delta t^3), \end{aligned}$$

where the last equality is obtained by (B.4). Inserting this and (B.4) into (B.7), we arrive at

$$w_j^{n,p,(0)} = w_j^{n,(0)} + \frac{p}{M}(w_j^{n,(1)} - w_j^{n,(0)}) + \frac{p^2}{M^2}\left(2 w_j^{n,(2)} - w_j^{n,(0)} - w_j^{n,(1)})\right) + O(\Delta t^3)$$

for $p = 0, 1, \ldots, M-1$. Similarly, we can approximate $w_j^{n,p,(1)}$ and $w_j^{n,p,(2)}$, the solutions at stage 1 and stage 2 of SSP-RK(3,3) as follows:

$$\begin{aligned} w_j^{n,p,(1)} &= w_j^{n,p,(0)} + \frac{\Delta t}{M} L_j(\mathbf{w}_j^{n,p,(0)}) \\ &= w_j^{n,p,(0)} + \frac{\Delta t}{M}\left(L_j(\mathbf{w}^{n,(0)}) + \nabla L_j(\mathbf{w}^{n,(0)}) \cdot (\mathbf{w}^{n,p,(0)} - \mathbf{w}^{n,(0)}) + O(\Delta t^2)\right) \\ &= w_j^{n,p,(0)} + \frac{\Delta t}{M} L_j(\mathbf{w}_j^{n,(0)}) + \frac{\Delta t}{M} \nabla L_j(\mathbf{w}^{n,(0)}) \cdot \frac{p \Delta t}{M} \partial_t \mathbf{w}^{n,(0)} + O(\Delta t^3) \\ &= w_j^{n,(0)} + \frac{p+1}{M}(w_j^{n,(1)} - w_j^{n,(0)}) + \frac{p(p+2)}{M^2}\left(2 w_j^{n,(2)} - w_j^{n,(0)} - w_j^{n,(1)})\right) + O(\Delta t^3), \end{aligned}$$

and

$$\begin{aligned} w_j^{n,p,(2)} &= \frac{3}{4} w_j^{n,p,(0)} + \frac{1}{4} w_j^{n,p,(1)} + \frac{1}{4} \frac{\Delta t}{M} L_j(\mathbf{w}^{n,p,(1)}) \\ &= \frac{3}{4} w_j^{n,p,(0)} + \frac{1}{4} w_j^{n,p,(1)} + \frac{1}{4} \frac{\Delta t}{M}\Big(L_j(\mathbf{w}^{n,(0)}) + \nabla L_j(\mathbf{w}^{n,(0)}) \cdot (\mathbf{w}^{n,p,(1)} - \mathbf{w}^{n,(0)}) \\ &\qquad + O(\Delta t^2)\Big) \\ &= w_j^{n,p,(0)} + \frac{\Delta t}{M} L_j(\mathbf{w}^{n,(0)}) + \frac{\Delta t}{M} \nabla L_j(\mathbf{w}^{n,(0)}) \cdot \frac{(p+1)\Delta t}{M} \partial_t \mathbf{w}^{n,(0)} + O(\Delta t^3) \\ &= w_j^{n,(0)} + \frac{2p+1}{2M}(w_j^{n,(1)} - w_j^{n,(0)}) + \frac{2p^2 + 2p + 1}{2M^2}\left(2 w_j^{n,(2)} - w_j^{n,(0)} - w_j^{n,(1)})\right) \\ &\qquad + O(\Delta t^3). \end{aligned}$$

B.3. **Predictor for the SSP-RK(5,4) scheme.** Again, we approximate $w_j^{n,p,(0)}$ by truncating (B.2), now to the fourth term:

(B.10)
$$w_j^{n,p,(0)} = w_j^{n,(0)} + \frac{p\Delta t}{M}\partial_t w_j^{n,(0)} + \frac{1}{2}\left(\frac{p\Delta t}{M}\right)^2 \partial_{tt} w_j^{n,(0)} + \frac{1}{6}\left(\frac{p\Delta t}{M}\right)^3 \partial_{ttt} w_j^{n,(0)} + O(\Delta t_p^4),$$

for $p = 0, 1, \ldots, M-1$. As for the second and third order cases, we approximate the time derivatives

$$\partial_t w_j^{n,(0)} = L_j(\mathbf{w}^{n,(0)}), \quad \partial_{tt} w_j^{n,(0)} = \nabla L_j(\mathbf{w}^{n,(0)}) \cdot \partial_t \mathbf{w}^{n,(0)}, \tag{B.11}$$

$$\partial_{ttt} w_j^{n,(0)} = (\partial_t \mathbf{w}^{n,(0)})^T \, \mathbf{H}_{L_j}(\mathbf{w}^{n,(0)}) \, \partial_t \mathbf{w}^{n,(0)} + \nabla L_j(\mathbf{w}^{n,(0)}) \cdot \partial_{tt} \mathbf{w}^{n,(0)}, \tag{B.12}$$

by using the solution the first four stages of SSP-RK(5,4) with the coarse time step size:

$$w_j^{n,(1)} = \alpha_{10} w_j^{n,(0)} + \beta_{10} \Delta t L_j(\mathbf{w}^{n,(0)}), \tag{B.13}$$

$$w_j^{n,(2)} = \alpha_{20} w_j^{n,(0)} + \alpha_{21} w_j^{n,(1)} + \beta_{21} \Delta t L_j(\mathbf{w}^{n,(1)}), \tag{B.14}$$

$$w_j^{n,(3)} = \alpha_{30} w_j^{n,(0)} + \beta_{30} \Delta t L_j(\mathbf{w}^{n,(0)}) + \alpha_{32} w_j^{n,(2)} + \beta_{32} \Delta t L_j(\mathbf{w}^{n,(2)}), \tag{B.15}$$

$$\begin{aligned} w_j^{n,(4)} = {} & \alpha_{40} w_j^{n,(0)} + \beta_{40} \Delta t L_j(\mathbf{w}^{n,(0)}) + \alpha_{41} w_j^{n,(1)} + \beta_{41} \Delta t L_j(\mathbf{w}^{n,(1)}) \\ & + \alpha_{43} w_j^{n,(3)} + \beta_{43} \Delta t L_j(\mathbf{w}^{n,(3)}). \end{aligned} \tag{B.16}$$

Denote by

$$\Delta t^{n,(i)} = \gamma^{(i)} \Delta t, \ i = 0, 1, 2, 3, \tag{B.17}$$

with

$$\gamma^{(0)} = 0, \ \gamma^{(1)} = \beta_{10}, \ \gamma^{(2)} = \alpha_{21}\gamma^{(1)} + \beta_{21}, \ \gamma^{(3)} = \alpha_{32}\gamma^{(2)} + \beta_{32} + \beta_{30}. \tag{B.18}$$

From (B.13) we have

$$\Delta t \, \partial_t w_j^{n,(0)} = \Delta t \, L_j(\mathbf{w}^{n,(0)}) = \frac{1}{\beta_{10}} \left(w_j^{n,(1)} - \alpha_{10} w_j^{n,(0)} \right).$$

Next, we approximate the flux by Taylor expansion with $O(\Delta t^3)$ truncated error:

(B.19)
$$\begin{aligned} L_j(\mathbf{w}^{n,(1)}) &= L_j(\mathbf{w}^{n,(0)}) + \nabla L_j(\mathbf{w}^{n,(0)}) \cdot (\mathbf{w}^{n,(1)} - \mathbf{w}^{n,(0)}) \\ &\quad + \frac{1}{2} (\mathbf{w}^{n,(1)} - \mathbf{w}^{n,(0)}) \, \mathbf{H}_{L_j}(\mathbf{w}^{n,(0)}) \, (\mathbf{w}^{n,(1)} - \mathbf{w}^{n,(0)}) + O(\Delta t^3) \\ &= L_j(\mathbf{w}^{n,(0)}) + \nabla L_j(\mathbf{w}^{n,(0)}) \cdot \left(\Delta t^{n,(1)} \, \partial_t \mathbf{w}^{n,(0)} + \frac{(\Delta t^{n,(1)})^2}{2} \, \partial_{tt} \mathbf{w}^{n,(0)} \right) \\ &\quad + \frac{(\Delta t^{n,(1)})^2}{2} \, (\partial_t \mathbf{w}^{n,(0)})^T \, \mathbf{H}_{L_j}(\mathbf{w}^{n,(0)}) \, \partial_t \mathbf{w}^{n,(0)} + O(\Delta t^3), \\ &= L_j(\mathbf{w}^{n,(0)}) + \Delta t^{n,(1)} \, \partial_{tt} w_j^{n,(0)} + \frac{(\Delta t^{n,(1)})^2}{2} \, \partial_{ttt} w_j^{n,(0)} + O(\Delta t^3), \end{aligned}$$

in which $\Delta t^{n,(1)}$ is defined in (B.17) and the last equality is obtained by substituting the derivatives in time (B.11)-(B.12). Similarly,

(B.20)
$$L_j(\mathbf{w}^{n,(2)}) = L_j(\mathbf{w}^{n,(0)}) + \Delta t^{n,(2)}\,\partial_{tt} w_j^{n,(0)} + \frac{\left(\Delta t^{n,(2)}\right)^2}{2}\,\partial_{ttt} w_j^{n,(0)} + O(\Delta t^3)$$

and

(B.21)
$$L_j(\mathbf{w}^{n,(3)}) = L_j(\mathbf{w}^{n,(0)}) + \Delta t^{n,(3)}\,\partial_{tt} w_j^{n,(0)} + \frac{\left(\Delta t^{n,(3)}\right)^2}{2}\,\partial_{ttt} w_j^{n,(0)} + O(\Delta t^3).$$

For the fourth order SSP-RK scheme, the number of stages is larger than the order of the scheme. Consequently, we can compute different approximations of the time derivatives $\partial_{tt} w_j^{n,(0)}$ and $\partial_{ttt} w_j^{n,(0)}$ using either equations (B.19)-(B.20) or (B.20)-(B.21). In particular, if we substitute the equations (B.19)-(B.20) into (B.14)-(B.15), we obtain the following system for $\partial_{tt} w_j^{n,(0)}$ and $\partial_{ttt} w_j^{n,(0)}$:

(B.22)
$$w_j^{n,(2)} = A_j + \beta_{21}\beta_{10}\Delta t^2 \partial_{tt} w_j^{n,(0)} + \beta_{21}\beta_{10}^2 \frac{\Delta t^3}{2} \partial_{ttt} w_j^{n,(0)} + O(\Delta t^4),$$

(B.23)
$$w_j^{n,(3)} = B_j + \beta_{32}(\alpha_{21}\beta_{10} + \beta_{21})\Delta t^2 \partial_{tt} w_j^{n,(0)} + \beta_{32}(\alpha_{21}\beta_{10} + \beta_{21})^2 \frac{\Delta t^3}{2} \partial_{ttt} w_j^{n,(0)} + O(\Delta t^4),$$

where

$$A_j = \alpha_{20} w_j^{n,(0)} + \alpha_{21} w_j^{n,(1)} + \frac{\beta_{21}}{\beta_{10}}\left(w_j^{n,(1)} - \alpha_{10} w_j^{n,(0)}\right),$$
$$B_j = \alpha_{30} w_j^{n,(0)} + \alpha_{32} w_j^{n,(2)} + \frac{(\beta_{30} + \beta_{32})}{\beta_{10}}\left(w_j^{n,(1)} - \alpha_{10} w_j^{n,(0)}\right).$$

By solving (B.22)-(B.23), we can compute fourth order approximations of $\partial_{tt} w_j^{n,(0)}$ and $\partial_{ttt} w_j^{n,(0)}$, denoted by $\partial_{tt} \overline{w}_j^{n,(0)}$ and $\partial_{ttt} \overline{w}_j^{n,(0)}$, as linear combinations of the solutions at different stages with the coarse time step size $w_j^{n,(0)}$, $w_j^{n,(1)}$, $w_j^{n,(2)}$, and $w_j^{n,(3)}$:

(B.24)
$$\begin{aligned} \partial_{tt} \overline{w}_j^{n,(0)} &= \partial_{tt} \overline{w}_j^{n,(0)}(w_j^{n,(0)}, w_j^{n,(1)}, w_j^{n,(2)}, w_j^{n,(3)}), \\ \partial_{ttt} \overline{w}_j^{n,(0)} &= \partial_{ttt} \overline{w}_j^{n,(0)}(w_j^{n,(0)}, w_j^{n,(1)}, w_j^{n,(2)}, w_j^{n,(3)}). \end{aligned}$$

Similarly, we can substitute the equations (B.20)-(B.21) into (B.15)-(B.16) to obtain alternative approximations of $\partial_{tt} w_j^{n,(0)}$ and $\partial_{ttt} w_j^{n,(0)}$, denoted by $\partial_{tt} \overline{\overline{w}}_j^{n,(0)}$ and $\partial_{ttt} \overline{\overline{w}}_j^{n,(0)}$, as linear combinations of the solutions $w_j^{n,(0)}$, $w_j^{n,(1)}$, $w_j^{n,(3)}$, and $w_j^{n,(4)}$:

(B.25)
$$\begin{aligned} \partial_{tt} \overline{\overline{w}}_j^{n,(0)} &= \partial_{tt} \overline{\overline{w}}_j^{n,(0)}(w_j^{n,(0)}, w_j^{n,(1)}, w_j^{n,(3)}, w_j^{n,(4)}), \\ \partial_{ttt} \overline{\overline{w}}_j^{n,(0)} &= \partial_{ttt} \overline{\overline{w}}_j^{n,(0)}(w_j^{n,(0)}, w_j^{n,(1)}, w_j^{n,(3)}, w_j^{n,(4)}). \end{aligned}$$

To take into account values at all four stages, we choose the average of (B.24) and (B.25) as approximations of $\partial_{tt} w_j^{n,(0)}$ and $\partial_{ttt} w_j^{n,(0)}$, respectively, and insert them into (B.10) to obtain a fourth order approximation of $w_j^{n,p,(0)}$.

Next, we approximate $w_j^{n,p,(i)}$, $i = 1, 2, 3, 4$ based on the definition of the SSP-RK(5,4) scheme:

$$w_j^{n,p,(i)} = \sum_{\nu=0}^{i-1} \alpha_{i\nu} w_j^{n,p,(\nu)} + \beta_{i\nu} \frac{\Delta t}{M} L_j(\mathbf{w}^{n,p,(\nu)}) \quad \forall\, i = 1, 2, 3, 4.$$

Denote by

$$\Delta t^{n,p,(i)} = (p + \gamma^{(i)}) \frac{\Delta t}{M},\ i = 0, 1, 2, 3,$$

with $\gamma^{(i)}$ defined in (B.18). We approximate $L_j(\mathbf{w}^{n,p,(i)})$, $i = 0, 1, 2, 3$, as in (B.19)-(B.21) and use (B.24) and (B.25) to approximate the time derivatives:

$$L_j(\mathbf{w}^{n,p,(i)}) = L_j(\mathbf{w}^{n,(0)}) + \Delta t^{n,p,(i)} \partial_{tt} w_j^{n,(0)} + \frac{1}{2} (\Delta t^{n,p,(i)})^2 \partial_{ttt} w_j^{n,(0)} + O\left(\Delta t^3\right).$$

Using this we can compute $w_j^{n,p,(i)}$, $i = 1, 2, 3, 4$ with fourth order accuracy in time.

Remark B.1. By construction, the predictors for SSP-RK(2, 2), SSP-RK(3, 3), and SSP-RK(5, 4) are, respectively, second, third, and fourth order accurate in time.

References

[1] A. Ashbourne, *Efficient Runge-Kutta Based Local Time-Stepping Methods*, Master Thesis, University of Waterloo, Canada, 2016.

[2] M. J. Berger and J. Oliger, *Adaptive mesh refinement for hyperbolic partial differential equations*, J. Comput. Phys. **53** (1984), no. 3, 484–512, DOI 10.1016/0021-9991(84)90073-1. MR739112

[3] M. J. Berger and R. J. Leveque, *Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems*, SIAM J. Numer. Anal. **35** (1998), no. 6, 2298–2316, DOI 10.1137/S0036142997315974. MR1655847

[4] G. Chavent and B. Cockburn, *The local projection P^0P^1-discontinuous-Galerkin finite element method for scalar conservation laws* (English, with French summary), RAIRO Modél. Math. Anal. Numér. **23** (1989), no. 4, 565–592, DOI 10.1051/m2an/1989230405651. MR1025072

[5] B. Cockburn, *Devising discontinuous Galerkin methods for non-linear hyperbolic conservation laws*, J. Comput. Appl. Math. **128** (2001), no. 1-2, 187–204, DOI 10.1016/S0377-0427(00)00512-4. Numerical analysis 2000, Vol. VII, Partial differential equations. MR1820874

[6] B. Cockburn and C.-W. Shu, *The Runge-Kutta local projection P^1-discontinuous-Galerkin finite element method for scalar conservation laws* (English, with French summary), RAIRO Modél. Math. Anal. Numér. **25** (1991), no. 3, 337–361, DOI 10.1051/m2an/1991250303371. MR1103092

[7] B. Cockburn and C.-W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework*, Math. Comp. **52** (1989), no. 186, 411–435, DOI 10.2307/2008474. MR983311

[8] B. Cockburn, S. Y. Lin, and C.-W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. III. One-dimensional systems*, J. Comput. Phys. **84** (1989), no. 1, 90–113, DOI 10.1016/0021-9991(89)90183-6. MR1015355

[9] B. Cockburn, S. Hou, and C.-W. Shu, *The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. The multidimensional case*, Math. Comp. **54** (1990), no. 190, 545–581, DOI 10.2307/2008501. MR1010597

[10] B. Cockburn and C.-W. Shu, *Runge-Kutta discontinuous Galerkin methods for convection-dominated problems*, J. Sci. Comput. **16** (2001), no. 3, 173–261, DOI 10.1023/A:1012873910884. MR1873283

[11] E. M. Constantinescu and A. Sandu, *Multirate timestepping methods for hyperbolic conservation laws*, J. Sci. Comput. **33** (2007), no. 3, 239–278, DOI 10.1007/s10915-007-9151-y. MR2357410

[12] C. Dawson, *High resolution upwind-mixed finite element methods for advection-diffusion equations with variable time-stepping*, Numer. Methods Partial Differential Equations **11** (1995), no. 5, 525–538, DOI 10.1002/num.1690110508. MR1345758

[13] C. Dawson and R. Kirby, *High resolution schemes for conservation laws with locally varying time steps*, SIAM J. Sci. Comput. **22** (2000), no. 6, 2256–2281, DOI 10.1137/S1064827500367737. MR1856633

[14] M. O. Domingues, S. M. Gomes, O. Roussel, and K. Schneider, *An adaptive multiresolution scheme with local time stepping for evolutionary PDEs*, J. Comput. Phys. **227** (2008), no. 8, 3758–3780, DOI 10.1016/j.jcp.2007.11.046. MR2403866

[15] M. Dumbser, M. Käser, and E. F. Toro, *An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes - V. Local time stepping and p-adaptivity*, Geophysical Journal International **171** (2007), 695-717.

[16] M. Dumbser, *Arbitrary-Lagrangian-Eulerian ADER-WENO finite volume schemes with time-accurate local time stepping for hyperbolic conservation laws*, Comput. Methods Appl. Mech. Engrg. **280** (2014), 57–83, DOI 10.1016/j.cma.2014.07.019. MR3255532

[17] R. E. Ewing and H. Wang, *A summary of numerical methods for time-dependent advection-dominated partial differential equations*, J. Comput. Appl. Math. **128** (2001), no. 1-2, 423–445, DOI 10.1016/S0377-0427(00)00522-7. Numerical analysis 2000, Vol. VII, Partial differential equations. MR1820884

[18] J. E. Flaherty, R. M. Loy, M. S. Shephard, B. K. Szymanski, J. D. Teresco, and L. H. Ziantz, *Adaptive Local Refinement with Octree Load Balancing for the Parallel Solution of Three-Dimensional Conservation Laws*, Journal of Parallel and Distributed Computing **47**(2) (1997), 139-152.

[19] M. J. Gander and L. Halpern, *Techniques for locally adaptive time stepping developed over the last two decades*, Domain Decomposition Methods in Science and Engineering XX, Lect. Notes Comput. Sci. Eng. **91**, Springer, Berlin (2007), 377-385.

[20] S. Gottlieb and C.-W. Shu, *Total variation diminishing Runge-Kutta schemes*, Math. Comp. **67** (1998), no. 221, 73–85, DOI 10.1090/S0025-5718-98-00913-2. MR1443118

[21] M. J. Grote, M. Mehlin, and T. Mitkova, *Runge-Kutta-based explicit local time-stepping methods for wave propagation*, SIAM J. Sci. Comput. **37** (2015), no. 2, A747–A775, DOI 10.1137/140958293. MR3324978

[22] S. Gupta, B. Wohlmuth, and R. Helmig, *Multi-rate timestepping schemes for hydro-geomechanical model for subsurface methane hydrate reservoirs*, Adv. Water Resour. **91** (2016), 78-87.

[23] A. Harten, *On a class of high resolution total-variation-stable finite-difference schemes*, SIAM J. Numer. Anal. **21** (1984), no. 1, 1–23, DOI 10.1137/0721001. With an appendix by Peter D. Lax. MR731210

[24] A. Harten, *Preliminary results on the extension of ENO schemes to two-dimensional problems*, Nonlinear hyperbolic problems (St. Etienne, 1986), Lecture Notes in Math., vol. 1270, Springer, Berlin, 1987, pp. 23–40, DOI 10.1007/BFb0078315. MR910102

[25] A. Harten and S. Osher, *Uniformly high-order accurate nonoscillatory schemes. I*, SIAM J. Numer. Anal. **24** (1987), no. 2, 279–309, DOI 10.1137/0724022. MR881365

[26] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy, *Uniformly high-order accurate essentially nonoscillatory schemes. III*, J. Comput. Phys. **71** (1987), no. 2, 231–303, DOI 10.1016/0021-9991(87)90031-3. MR897244

[27] T.-T.-P. Hoang, W. Leng, L. Ju, Z. Wang, and K. Pieper, *Conservative explicit local time-stepping schemes for the shallow water equations*, J. Comput. Phys. **382** (2019), 152–176, DOI 10.1016/j.jcp.2019.01.006. MR3905215

[28] A. N. Brooks and T. J. R. Hughes, *Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations*, Comput. Methods Appl. Mech. Engrg. **32** (1982), no. 1-3, 199–259, DOI 10.1016/0045-7825(82)90071-8. FENOMECH "81, Part I (Stuttgart, 1981). MR679322

[29] G.-S. Jiang and C.-W. Shu, *Efficient implementation of weighted ENO schemes*, J. Comput. Phys. **126** (1996), no. 1, 202–228, DOI 10.1006/jcph.1996.0130. MR1391627

[30] C. Johnson, A. Szepessy, and P. Hansbo, *On the convergence of shock-capturing streamline diffusion finite element methods for hyperbolic conservation laws*, Math. Comp. **54** (1990), no. 189, 107–129, DOI 10.2307/2008684. MR995210

[31] L. Krivodonova, *An efficient local time-stepping scheme for solution of nonlinear conservation laws*, J. Comput. Phys. **229** (2010), no. 22, 8537–8551, DOI 10.1016/j.jcp.2010.07.037. MR2719187

[32] E. J. Kubatko, B. A. Yeager, and D. I. Ketcheson, *Optimal strong-stability-preserving Runge-Kutta time discretizations for discontinuous Galerkin methods*, J. Sci. Comput. **60** (2014), no. 2, 313–344, DOI 10.1007/s10915-013-9796-7. MR3225785

[33] P. D. Lax, *Weak solutions of nonlinear hyperbolic equations and their numerical computation*, Comm. Pure Appl. Math. **7** (1954), 159–193, DOI 10.1002/cpa.3160070112. MR66040

[34] X.-D. Liu, S. Osher, and T. Chan, *Weighted essentially non-oscillatory schemes*, J. Comput. Phys. **115** (1994), no. 1, 200–212, DOI 10.1006/jcph.1994.1187. MR1300340

[35] F. Lörcher, G. Gassner, and C.-D. Munz, *A discontinuous Galerkin scheme based on a space-time expansion. I. Inviscid compressible flow in one space dimension*, J. Sci. Comput. **32** (2007), no. 2, 175–199, DOI 10.1007/s10915-007-9128-x. MR2320569

[36] K. W. Morton and P. K. Sweby, *A comparison of flux limited difference methods and characteristic Galerkin methods for shock modelling*, J. Comput. Phys. **73** (1987), 203-230.

[37] S. Osher, *Convergence of generalized MUSCL schemes*, SIAM J. Numer. Anal. **22** (1985), no. 5, 947–961, DOI 10.1137/0722057. MR799122

[38] S. Osher and S. Chakravarthy, *High resolution schemes and the entropy condition*, SIAM J. Numer. Anal. **21** (1984), no. 5, 955–984, DOI 10.1137/0721060. MR760626

[39] S. Osher and R. Sanders, *Numerical approximations to nonlinear conservation laws with locally varying time and space grids*, Math. Comp. **41** (1983), no. 164, 321–336, DOI 10.2307/2007679. MR717689

[40] S. Osher and E. Tadmor, *On the convergence of difference approximations to scalar conservation laws*, Math. Comp. **50** (1988), no. 181, 19–51, DOI 10.2307/2007913. MR917817

[41] T. D. Ringler, J. Thuburn, J. B. Klemp, and W. C. Skamarock, *A unified approach to energy conservation and potential vorticity dynamics for arbitrarily-structured C-grids*, J. Comput. Phys. **229** (2010), no. 9, 3065–3090, DOI 10.1016/j.jcp.2009.12.007. MR2601090

[42] I. Rybak, J. Magiera, R. Helmig, and C. Rohde, *Multirate time integration for coupled saturated/unsaturated porous medium and free flow systems*, Comput. Geosci. **19** (2015), no. 2, 299–309, DOI 10.1007/s10596-015-9469-8. MR3351781

[43] B. F. Sanders, *Integration of a shallow water model with a local time-step*, J. Hydraul. Res. **46** (4) (2008), 466-475.

[44] A. Sandu and E. M. Constantinescu, *Multirate explicit Adams methods for time integration of conservation laws*, J. Sci. Comput. **38** (2009), no. 2, 229–249, DOI 10.1007/s10915-008-9235-3. MR2471015

[45] C.-W. Shu, *TVB uniformly high-order schemes for conservation laws*, Math. Comp. **49** (1987), no. 179, 105–121, DOI 10.2307/2008252. MR890256

[46] C.-W. Shu, *TVB boundary treatment for numerical solutions of conservation laws*, Math. Comp. **49** (1987), no. 179, 123–134, DOI 10.2307/2008253. MR890257

[47] C.-W. Shu, *Total-variation-diminishing time discretizations*, SIAM J. Sci. Statist. Comput. **9** (1988), no. 6, 1073–1084, DOI 10.1137/0909073. MR963855

[48] C.-W. Shu and S. Osher, *Efficient implementation of essentially nonoscillatory shock-capturing schemes*, J. Comput. Phys. **77** (1988), no. 2, 439–471, DOI 10.1016/0021-9991(88)90177-5. MR954915

[49] C.-W. Shu and S. Osher, *Efficient implementation of essentially nonoscillatory shock-capturing schemes. II*, J. Comput. Phys. **83** (1989), no. 1, 32–78, DOI 10.1016/0021-9991(89)90222-2. MR1010162

[50] G. A. Sod, *A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws*, J. Computational Phys. **27** (1978), no. 1, 1–31, DOI 10.1016/0021-9991(78)90023-2. MR0495002

[51] P. K. Sweby, *High resolution schemes using flux limiters for hyperbolic conservation laws*, SIAM J. Numer. Anal. **21** (1984), no. 5, 995–1011, DOI 10.1137/0721062. MR760628

[52] A. Taube, M. Dumbser, C. Munz and R. Schneider, *A high-order discontinuous Galerkin method with time-accurate local time stepping for the Maxwell equations*, Int. J. Numer. Model. **22**, 2009, pp. 77-103.

[53] J. Thuburn, T. D. Ringler, W. C. Skamarock, and J. B. Klemp, *Numerical representation of geostrophic modes on arbitrarily structured C-grids*, J. Comput. Phys. **228** (2009), no. 22, 8321–8335, DOI 10.1016/j.jcp.2009.08.006. MR2574097

[54] V. A. Titarev and E. F. Toro, *ADER: arbitrary high order Godunov approach*, Proceedings of the Fifth International Conference on Spectral and High Order Methods (ICOSAHOM-01) (Uppsala), J. Sci. Comput. **17** (2002), no. 1-4, 609–618, DOI 10.1023/A:1015126814947. MR1910755

[55] C. J. Trahan and C. Dawson, *Local time-stepping in Runge-Kutta discontinuous Galerkin finite element methods applied to the shallow-water equations*, Comput. Methods Appl. Mech. Engrg. **217/220** (2012), 139–152, DOI 10.1016/j.cma.2012.01.002. MR2897818

[56] B. Van Leer, *Towards the ultimate conservative difference scheme, V. A second order sequel to Godunov's method*, J. Comput. Phys. **32** (1979), 101-136.

[57] P. Woodward and P. Colella, *The numerical simulation of two-dimensional fluid flow with strong shocks*, J. Comput. Phys. **54** (1984), no. 1, 115–173, DOI 10.1016/0021-9991(84)90142-6. MR748569

Department of Mathematics and Statistics, Auburn University, Auburn, Alabama 36849

Email address: tzh0059@auburn.edu

Department of Mathematics, University of South Carolina, Columbia, South Carolina 29208

Email address: ju@math.sc.edu

State Key Laboratory of Scientific and Engineering Computing, Chinese Academy of Sciences, Beijing 100190, People's Republic of China

Email address: wleng@lsec.cc.ac.cn

Department of Mathematics, University of South Carolina, Columbia, South Carolina 29208

Email address: wangzhu@math.sc.edu