MAPPED TENT PITCHING SCHEMES FOR HYPERBOLIC SYSTEMS *

J. GOPALAKRISHNAN †, J. SCHÖBERL ‡, AND C. WINTERSTEIGER †

Abstract. A spacetime domain can be progressively meshed by tent shaped objects. Numerical methods for solving hyperbolic systems using such tent meshes to advance in time have been proposed previously. Such schemes have the ability to advance in time by different amounts at different spatial locations. This paper explores a technique by which standard discretizations, including explicit time stepping, can be used within tent-shaped spacetime domains. The technique transforms the equations within a spacetime tent to a domain where space and time are separable. After detailing techniques based on this mapping, several examples including the acoustic wave equation and the Euler system are considered.

Key words. local time stepping, wave, causality, Piola, entropy residual, gas dynamics

AMS subject classifications. 65M60, 65M20

1. Introduction.

We introduce a new class of methods called Mapped Tent Pitching (MTP) schemes for numerically solving hyperbolic problems. These schemes can be thought of as fully explicit or locally implicit schemes on unstructured spacetime meshes obtained by a process known in the literature as tent pitching. This process creates an advancing front in spacetime made by canopies of tent-shaped regions. Spacetime tents are erected (with time as the last or vertical dimension in spacetime – see Figure 1) so that causality constraints of the hyperbolic problem are never violated and the hyperbolic problem is solved progressively in layers of tents. Such meshing processes were named tent pitching in [4, 22]. In this paper, we will refer to tent pitching as a discretization scheme together with all the attendant meshing techniques. In fact, the main focus of this paper is not on meshing, but rather on novel discretization techniques that exploit tent pitched meshes.

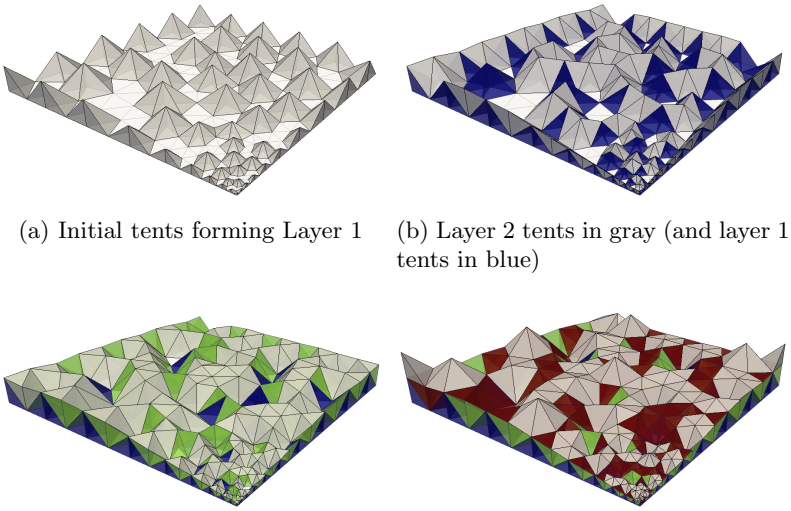
Today, the dominant discretization technique that use tent pitched meshes is the spacetime discontinuous Galerkin (SDG) method. Its origins can be traced back to [12, 19]. It has seen active development over the years in engineering applications [14, 18, 27] and has also motivated several works in numerical analysis [5, 7, 15]. The SDG schemes use piecewise polynomials in the spacetime elements (with no continuity constraints across mesh element interfaces) and a DG (discontinuous Galerkin) style spacetime discretization. Different prescriptions of DG fluxes result in different methods. Advanced techniques, including adaptive spacetime mesh refinement maintaining causality [16], and exact conservation [1], have been realized for SDG methods.

The above-mentioned research into SDG methods has abundantly clarified the many advantages that tent pitched meshes offer. Perhaps the primary advantage they offer is a rational way to build high order methods (in space and time) that incorporates spatial adaptivity and locally varying time step size, even on complex structures. Without tent meshes, many standard methods resort to ad hoc techniques (interpolation, extrapolation, projection, etc.) for locally adaptive time stepping [6] within inexpensive explicit strategies. If one is willing to pay the expense of solving

*This work was supported in part by the NSF under DMS-1318916 and DMS-1624776.

†Portland State University, PO Box 751, Portland OR 97207, USA (gjay@pdx.edu).

‡Wiedner Hauptstraße 8-10, Technische Universität Wien, 1040 Wien, Austria (joachim.schoeberl@tuwien.ac.at, christoph.wintersteiger@tuwien.ac.at).



(a) Initial tents forming Layer 1

(b) Layer 2 tents in gray (and layer 1 tents in blue)

(c) Layer 3 tents in gray (and tents of previous layers in other colors)

(d) Layer 4 tents (in gray)

Fig. 1: Parallel tents within different layers

global systems on spacetime domains [17, 23, 24], then time and space adaptivity are easy. In between these options, there are interesting alternative methods, without using tents, able to perform explicit local time stepping while maintaining high order accuracy [3, 8] by dividing the spatial mesh into fine and coarse regions. The concepts we present using tents provide a different avenue for locally advancing in time. Recall that when solving hyperbolic systems on a spacetime domain, we must ensure that the domain of dependence of all points are contained within the domain. Tents provide a natural way to ensure this by restricting the height of the tent pole. This height restriction is referred to as the *causality constraint* and it restricts the maximal time advance possible at a spatial point. Even if akin to the Courant-Friedrichs-Levy (CFL) constraint, the causality constraint does not arise from a discretization and is different from the CFL constraint.

The main novelty in MTP schemes is a mapping of tents to cylindrical domains where space and time can be separated, so that standard spatial discretizations combined with time stepping can be used for solving on each tent. MTP schemes proceed as follows:

1. Construct a spacetime mesh using an advancing front tent meshing algorithm (see section 2).
2. Map the hyperbolic system on each tent to a system on a spacetime cylinder (see Theorem 2).
3. Apply a method of lines discretization within the spacetime cylinder, i.e., use any appropriate (high order) spatial discretization and combine with (multiple) explicit or implicit time steps within the cylinder.
4. Map the computed solution on the cylinder back to the tent.

Proceeding along these steps, tent by tent, we obtain the entire spacetime solution. Note that high order methods are obtained by increasing the polynomial degree of

the spatial approximation and correspondingly increasing the number of time steps within each mapped tent.

All tent pitching schemes advance the solution tent by tent without global matrix inversions. However, for the first time, as a result of the above-mentioned mapping strategy, we are able to construct tent pitching schemes that are explicit not only from tent to tent, but also within each tent. We call these *explicit MTP schemes*. (Note that the possibility to perform explicit time stepping within a tent did not exist with SDG methods.) Explicit MTP schemes map each tent to a cylinder, where space and time can be separated, use a spatial discretization and thereafter apply an explicit time stepping to compute the tent solution. Using explicit MTP schemes, we are able to bring the well-known cache-friendliness and data locality properties of explicit methods into the world of local time stepping through unstructured spacetime tent meshes. In a later section, we will show the utility of explicit MTP schemes by applying it to a complex Mach 3 wind tunnel problem using an existing DG discretization in space and an explicit time stepping. Note that there is no need to develop a new spacetime formulation on tents for the Euler system in order to apply the MTP scheme.

The new mapping strategy also permits the creation of another class of novel methods which we call *locally implicit MTP schemes*. Here, after the mapping each tent to a cylinder, we use an implicit time stepping algorithm. This requires us to solve a small spatial system (local to the tent) in order to advance the hyperbolic solution on each tent. This approach also retains the advantage of being able to use standard existing spatial discretizations and well-known high order implicit Runge-Kutta time stepping. While the explicit MTP schemes are constrained by both the causality constraint and a CFL constraint imposed by the choice of the spatial discretization, in locally implicit MTP schemes there is no CFL constraint. The causality constraint applies, and depends on the *local* tent geometry and local wavespeed, but is independent of degree (p) of the spatial discretization. This provides one point of contrast against traditional methods, whose global timestep ($h_{\min}/p_{\max}^2$) depends on the smallest element size ($h_{\min}$) and the largest degree ($p_{\max}$) over the entire mesh.

In the remainder of the paper, we will be concerned with hyperbolic problems that fit into a generic definition described next. Let L and N be integers not less than 1. All the problems considered can be written as a system of L equations on a spacetime cylindrical domain $\Omega = \Omega_0 \times (0, t_{\max})$, where the spatial domain Ω_0 is contained in $\mathbb{R}^N$. Given sufficiently regular functions $f : \Omega \times \mathbb{R}^L \to \mathbb{R}^{L\times N}$, $g : \Omega \times \mathbb{R}^L \to \mathbb{R}^L$, and $b : \Omega \times \mathbb{R}^L \to \mathbb{R}^L$, the problem is to find a function $u : \Omega \to \mathbb{R}^L$ satisfying

$$\partial_t g(x,t,u) + \operatorname{div}_x f(x,t,u) + b(x,t,u) = 0 \tag{1.1}$$

where $\partial_t = \partial/\partial t$ denotes the time derivative and $\operatorname{div}_x(\cdot)$ denotes the spatial divergence operator applied row wise to matrix-valued functions. To be clear, the system (1.1) can be rewritten, using subscripts to denote components (e.g., b_l denotes the lth component of b, f_{li} denotes the (l,i)th component of f, etc.), as

$$\partial_t g_l(x,t,u(x,t)) + \sum_{i=1}^{N} \partial_i (f_{li}(x,t,u(x,t)) + b_l(x,t,u(x,t)) = 0, \tag{1.2}$$

for $l = 1, \ldots, L$. Here and throughout, $\partial_i = \partial/\partial x_i$ denotes differentiation along the ith direction in $\mathbb{R}^N$. In examples, we will supplement (1.2) by initial conditions on Ω_0 and boundary conditions on $\partial\Omega_0 \times (0, t_{\max})$.

We assume that the system (1.1) is *hyperbolic in the t-direction*, as defined in [2]. Note that in particular, this requires that for any fixed x, t, and u, the $L \times L$ derivative matrix $D_u g$ (whose (l, m)th entry is $\partial g_m / \partial u_l$) is invertible, i.e.,

$$\det[D_u g] \neq 0. \tag{1.3}$$

Hyperbolicity also provides, for each direction vector and each point x, t, u, a series of real eigenvalues called characteristic speeds. Let $c(x, t, u)$ denote the maximum of these speeds for all directions. For simplicity, we assume that $c(x, t, u)$ is given (even though it can often be computationally estimated), so that the meshing process in the next section can use it as input.

Geometrical definitions and meshing algorithms are given in Section 2 (Tents). Transformation of tents and hyperbolic equations within them is the subject of Section 3 (Maps). Two distinct approaches to designing novel MTP methods are presented in Section 4. In Section 5, we discuss a locally implicit MTP method for the acoustic wave equation in detail. In Section 6, after giving general details pertaining to treatment of nonlinear hyperbolic conservation laws, we focus on an explicit MTP scheme for Euler equations.

2. Tents. The MTP schemes we present in later sections fall into the category of methods that use tent pitching for unstructured spacetime meshing. Accordingly, in this section, we first give a general description of tent meshing, clarifying the mathematical meaning of words we have already used colloquially such as "tent," "tent pole," "advancing front," etc., and then give details of a specific meshing algorithm that we have chosen to implement.

2.1. Overview of a tent pitching scheme. We now describe how a tent pitching scheme advances the numerical solution in time. We mesh Ω_0 by a simplicial conforming shape regular finite element mesh $\mathcal{T}$. The mesh is unstructured to accomodate for any intricate features in the spatial geometry or in the evolving solution. Let $P_1(\mathcal{T})$ denote the set of continuous real-valued functions on Ω_0 which are linear on each element of $\mathcal{T}$. Clearly any function in $P_1(\mathcal{T})$ is completely determined by its values at the vertices v_l, $l = 1, \ldots, N_{\mathcal{T}}$, of the mesh $\mathcal{T}$.

At the ith step of a tent pitching scheme, the numerical solution is available for all $x \in \Omega_0$ and all $0 < t < \tau_i(x)$. The function τ_i is in $P_1(\mathcal{T})$. The graph of τ_i, denoted by S_i, and is called the "advancing front" (see Figure 1.) We present a serial version of the algorithm first. A parallel generalization is straightforward as mentioned in Remark 1. A tent pitching scheme updates τ_i within the general outline of Algorithm 1.

The height of the tent pole k_i at each step should be determined using the causality constraint so that (1.1) is solvable on K_i. The choice of the vertex $v^{(i)}$ should be made considering the height of the neighboring vertices. Other authors have studied these issues [4, 22] and given appropriate advancing front meshing strategies. Next, we describe a simple strategy which we have chosen to implement. It applies verbatim in both two and three space dimensions.

2.2. Algorithm to mesh by tents. To motivate our meshing strategy, first let $\bar{c}(x)$ denote a given (or computed) approximation to the maximal characteristic speed at a point $(x, \tau_{i-1}(x))$ on the advancing front S_{i-1}, e.g., $\bar{c}(x) = c(x, \tau_{i-1}(x), u(x, \tau_{i-1}(x))$, where u is the computed numerical solution. We want to ensure that,

Algorithm 1 Advancing front of tents and approximate solution

1. Initially, set $\tau_0 \equiv 0$. Then $S_0 = \Omega_0$. The solution on S_0 is determined by the initial data on Ω_0.
2. For $i = 1, 2, \ldots,$ do:
   (a) Find a mesh vertex $v^{(i)}$ where good relative progress in time can be made and calculate the height (in time) k_i by which we can move the advancing front at v_i. One strategy to do this is detailed below in Algorithm 2.
   (b) Given the solution on the current advancing front S_{i-1}, pitch a "space-time tent" K_i by erecting a "tent pole" of height k_i at the point $(v^{(i)}, \tau_{i-1}(v^{(i)}))$ on S_{i-1}. Let $\eta_i \in P_1(\mathcal{T})$ be the unique function that equals one at $v^{(i)}$ and is zero at all other mesh vertices. Set

   $$\tau_i = \tau_{i-1} + k_i \eta_i \tag{2.1}$$

   Define the "vertex patch" Ω_v of a mesh vertex v as the (spatial) open set in $\mathbb{R}^N$ that is the interior of the union of all simplices in $\mathcal{T}$ connected to v. Then the tent K_i can be expressed as

   $$K_i = \{(x,t) : \ x \in \Omega_{v^{(i)}}, \ \tau_{i-1}(x) < t < \tau_i(x)\}. \tag{2.2}$$

   (c) Numerically solve (1.1) on K_i (e.g., by the methods proposed in the later sections of this paper). Initial data is obtained from the given solution on S_{i-1}. If $v^{(i)} \in \partial\Omega_0$, then the boundary conditions required to solve (1.1) on K_i are obtained from the given boundary conditions on the global boundary $\partial\Omega_0 \times (0, t_{\max})$.
   (d) If $\tau(v) \geq t_{\max}$ for all mesh vertices v, then exit.

for all $x \in \Omega_0$, we have

$$|\mathrm{grad}_x \tau_i(x)| \ < \ \frac{1}{\bar{c}(x)} \tag{2.3}$$

at every step i. Here $|\cdot|$ denotes the Euclidean norm. This is our *causality condition*, which is imposed even before we have discretized the hyperbolic problem. Note that this has also been called the "cone constraint" in [4], where it is geometrically interpreted as mesh facets separating the domain of influence (light cone opening above) from the domain of dependence (light cone opening below).

For simplicity, we now assume that c is independent of time and impose the following condition which is more stringent than (2.3):

$$\left|\mathrm{grad}_x(\tau_i|_T)\right| \ < \ \frac{1}{c_T}, \qquad \text{for all } T \in \mathcal{T}, \tag{2.4}$$

where $c_T = \max_{x \in T} \bar{c}(x)$. Since $\tau_i|_T$ is linear, its gradient is a constant vector that is determined by its tangential components along the edges of T. The tangential component on a mesh edge e of length $|e|$ is $(\tau_i(e_1) - \tau_i(e_2))/|e|$, where e_1 and e_2 denote the endpoints of e. Hence, by virtue of our assumption that the initial spatial mesh is shape regular, we can guarantee that (2.4) holds by imposing

$$\frac{\tau_i(e_1) - \tau_i(e_2)}{|e|} \leq \frac{C_{\mathcal{T}}}{c_e}, \qquad \text{for all mesh edges } e, \tag{2.5}$$

where c_e is the maximum of c_T over all elements T which have e as an edge and $C_{\mathcal{T}}$ is a constant that depends only on the shape regularity of the mesh $\mathcal{T}$. Condition (2.5) is easier to work with in practice and is the same in two and three space dimensions. A practical strategy is to start with a guess for $C_{\mathcal{T}}$ like $1/3$, check if the values of $\text{grad}_x \tau_i$ at the integration nodes (which need to be computed anyway as will be clear later) satisfy (2.4), and revise if necessary.

To obtain an advancing front satisfying (2.5) at all stages i, we maintain a list of potential time advance $\tilde{k}_l^{(i)}$ that can be made at any vertex v_l. Let $\mathcal{E}_l$ denote the set of all mesh edges connected to the vertex v_l and suppose edge endpoints are enumerated so that $e_1 = v_l$ for all $e \in \mathcal{E}_l$. Given τ_i satisfying (2.5), while considering pitching a tent at $(v_l, \tau_i(v_l))$ so that (2.5) continues to hold, we want to ensure that

$$\frac{\left(\tau_i(v_l) + \tilde{k}_l^{(i)}\right) - \tau_i(e_2)}{|e|} \leq \frac{C_{\mathcal{T}}}{c_e} \quad \text{and} \quad \frac{-\left(\tau_i(v_l) + \tilde{k}_l^{(i)}\right) + \tau_i(e_2)}{|e|} \leq \frac{C_{\mathcal{T}}}{c_e}$$

hold for all $e \in \mathcal{E}_l$. The latter inequality is obvious from (2.5) since we are only interested in $\tilde{k}_l^{(i)} \geq 0$. The former inequality is ensured if we choose

$$\tilde{k}_l^{(i)} \leq \min_{e \in \mathcal{E}_l} \left(\tau_i(e_2) - \tau_i(v_l) + |e| \frac{C_{\mathcal{T}}}{c_e} \right),$$

as done in the Algorithm 2 below. The algorithm also maintains a list of locations ready for pitching a tent. For this, it needs the reference heights $r_l = \min_{e \in \mathcal{E}_l} |e| C_{\mathcal{T}} / c_e$ (the maximal tent pole heights on a flat advancing front) which can be precomputed. Set $\tilde{k}_l^{(0)} = r_l$. A vertex v_l is considered a location where "good" progress in time can be made if its index l is in the set

$$J_i = \left\{ l : \ \tilde{k}_l^{(i)} \geq \gamma r_l \right\}. \tag{2.6}$$

Here $0 < \gamma < 1$ is a parameter (usually set to $1/2$). While a lower value of γ identifies many vertices to progress in time moderately, a higher value of γ identifies fewer vertices where time can be advanced more aggressively.

Algorithm 2 Updating potential pitch locations and time steps

Initially, $\tau_0 \equiv 0$, $\tilde{k}_l^{(0)} = r_l$ and $J_0 = \{1, 2 \cdots, N_{\mathcal{T}}\}$. For $i \geq 1$, given τ_{i-1}, $\{\tilde{k}_l^{(i-1)}\}$, and J_{i-1}, we choose the next tent pitching location $(v^{(i)})$ and the tent pole height (k_i), and update as follows:

1. Pick any l_* in J_{i-1}.
2. Set $v^{(i)} = v_{l_*}$ and $k_i = \tilde{k}_{l_*}^{(i-1)}$.
3. Update τ_i by (2.1).
4. Update $\tilde{k}_l^{(i)}$ for all vertices v_l adjacent to $v^{(i)}$ by

$$\tilde{k}_l^{(i)} = \min \left(t_{\max} - \tau_i(v_l), \ \min_{e \in \mathcal{E}_l} \left(\tau_i(e_2) - \tau_i(v_l) + |e| \frac{C_{\mathcal{T}}}{c_e} \right) \right).$$

5. Use $\{\tilde{k}_l^{(i)}\}$ to set J_i using (2.6).

Remark 1 (Parallel tent pitching). To pitch multiple tents in parallel, at the ith step, instead of picking l_* arbitrarily as in Algorithm 2, we choose $l_* \in J_{i-1}$ with

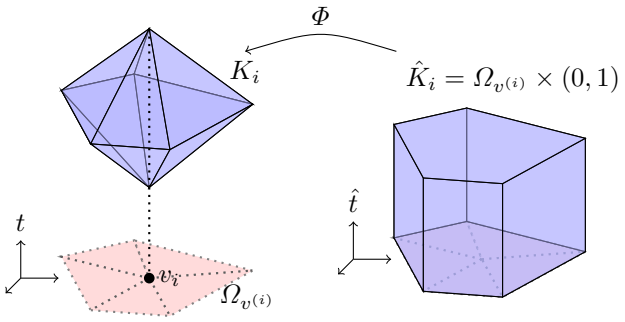


Fig. 2: Tent mapped from a tensor product domain.

the property that $\Omega_{v_{l_*}} = \Omega_{v^{(i)}}$ does not intersect $\Omega_{v^{(j)}}$ for all $j < i$. As we step through i, we continue to pick such l_* until we reach an index $i = i_1$ where no such l_* exists. All the tents made until this point, say $K_1, K_2, \ldots, K_{i_1}$ form the *layer* L_1. (An example of tents within such layers are shown in Figure 1 – in this example one of the corners of the domain has a singularity.) We then repeat this process to find greater indices $i_2 < i_3 < \cdots$ and layers $L_k = \{K_{i_{k-1}}, K_{i_{k-1}+1}, \ldots, K_{i_k}\}$ with the property that $\Omega_{v^{(j)}}$ does not intersect $\Omega_{v^{(i)}}$ for any distinct i and j in the range $i_{k-1} \le i, j \le i_k$. Computations on tents within each layer can proceed in parallel.

3. Maps. In this section we discuss a mapping technique that allows us to separate space and time discretizations within tents. Domains like $\Omega_0 \times (0,T)$ formed by a tensor product of a spatial domain with a time interval are referred to as spacetime cylinders. Such domains are amenable to tensor product discretizations where the space and time discretizations neatly separate. However, the tent K_i in (2.2) is not of this form. Therefore, we now introduce a mapping that transforms K_i one-to-one onto the spacetime cylinder $\hat{K}_i = \Omega_{v^{(i)}} \times (0,1)$.

Define the mapping $\Phi : \hat{K}_i \to K_i$ (see Figure 2) by $\Phi(\hat{x},\hat{t}) = (\hat{x}, \varphi(\hat{x},\hat{t}))$, where $\varphi(\hat{x},\hat{t}) = (1-\hat{t})\tau_{i-1}(\hat{x}) + \hat{t}\tau_i(\hat{x})$, for all $(\hat{x},\hat{t})$ in $\hat{K}_i$. Note that the $(N+1)\times(N+1)$ Jacobian matrix of derivatives of Φ takes the form

$$\hat{D}\Phi = \begin{bmatrix} I & 0 \\ \hat{D}\varphi & \delta \end{bmatrix} \tag{3.1}$$

where $\hat{D}\varphi = [\hat{\text{grad}}\,\varphi]^t = [\hat{\partial}_1\varphi\ \hat{\partial}_2\varphi\ \cdots\ \hat{\partial}_N\varphi]$, and $\delta = \tau_i - \tau_{i-1}$. Here and throughout, we use abbreviated notation for derivates $\hat{\partial}_j = \partial/\partial\hat{x}_j$, $\hat{\partial}_t = \partial/\partial\hat{t} = \hat{\partial}_{N+1}$ that also serves to distinguish differentiation on $\hat{K}_i$ from differentiation (∂_i) on K_i. Define

$$\hat{f}(\hat{x},\hat{t},w) = f(\Phi(\hat{x},\hat{t}),w), \qquad \hat{g}(\hat{x},\hat{t},w) = g(\Phi(\hat{x},\hat{t}),w), \tag{3.2a}$$

$$\hat{b}(\hat{x},\hat{t},w) = b(\Phi(\hat{x},\hat{t}),w), \qquad \hat{G}(\hat{x},\hat{t},w) = \hat{g}(\hat{x},\hat{t},w) - \hat{f}(\hat{x},\hat{t},w)\,\hat{\text{grad}}\,\varphi \tag{3.2b}$$

$$\hat{u} = u \circ \Phi, \qquad \hat{U}(\hat{x},\hat{t}) = \hat{G}(\hat{x},\hat{t},\hat{u}(\hat{x},\hat{t})). \tag{3.2c}$$

The last equation, showing that the function $\hat{u} : \hat{K}_i \to \mathbb{R}^L$ is mapped to $\hat{U} : \hat{K}_i \to \mathbb{R}^L$ by $\hat{G}$, will often be abbreviated as simply $\hat{U} = \hat{G}(\hat{u})$.

THEOREM 2. *The function u satisfies (1.1) in K_i if and only if $\hat{u}$ and $\hat{U}$ satisfy*

$$\hat{\partial}_t\hat{U} + \hat{\text{div}}\,(\delta\hat{f}\,) + \delta\,\hat{b} = 0 \qquad \text{in } \hat{K}_i,$$

which in component form reads as

$$\hat{\partial}_t[\hat{G}(\hat{u})]_l + \sum_{j=1}^{N} \hat{\partial}_j \left(\delta(\hat{x}) \hat{f}_{lj}(\hat{x}, \hat{t}, \hat{u}(\hat{x}, \hat{t}))\right) + \delta(\hat{x}) \hat{b}_l(\hat{x}, \hat{t}, \hat{u}(\hat{x}, \hat{t})) = 0 \tag{3.3}$$

for all $(\hat{x}, \hat{t})$ *in* $\hat{K}_i$ *and all* $l = 1, \ldots, L$.

Proof. The proof proceeds by calculating the pull back of the system (1.1) from K_i to $\hat{K}_i$ using the map Φ. Using the given u, define $F_l : K_i \to \mathbb{R}^{N+1}$ and $B : K_i \to \mathbb{R}^L$ by

$$F_l(x,t) = \begin{bmatrix} f_{l1}(x,t,u(x,t)) \\ \vdots \\ f_{lN}(x,t,u(x,t)) \\ g_l(x,t,u(x,t)) \end{bmatrix}, \quad B(x,t) = \begin{bmatrix} b_1(x,t,u(x,t)) \\ \vdots \\ b_L(x,t,u(x,t)) \end{bmatrix}$$

and define their pullbacks on $\hat{K}_i$ by

$$\hat{F}_l = \det[\hat{D}\Phi]\,[\hat{D}\Phi]^{-1}(F_l \circ \Phi), \qquad \hat{B} = \det[\hat{D}\Phi]\,(B \circ \Phi).$$

By the well-known properties of the Piola map,

$$\hat{\operatorname{div}} \hat{F} = \det[\hat{D}\Phi]\,(\operatorname{div} F) \circ \Phi, \tag{3.4}$$

where the divergence on either side is now taken in spacetime ($\mathbb{R}^{N+1}$). Note that $\det[\hat{D}\Phi] = \delta$ is never zero at any point of (the open set) $\hat{K}_i$. Writing equation (1.1) in these new notations, we obtain $(\operatorname{div} F_l)(x,t) + B(x,t) = 0$ for all $(x,t) \in K_i$, or equivalently,

$$(\operatorname{div} F_l)(\Phi(\hat{x}, \hat{t})) + B(\Phi(\hat{x}, \hat{t})) = 0$$

for all $(\hat{x}, \hat{t}) \in \hat{K}_i$. Multiplying through by $\det[\hat{D}\Phi]$ and using (3.4), this becomes

$$\hat{\operatorname{div}} \hat{F}_l + \hat{B} = 0, \qquad \text{on } \hat{K}_i. \tag{3.5}$$

To finish the proof, we simplify this equation. Inverting the block triangular matrix $\hat{D}\Phi$ displayed in (3.1) and using it in the definition for $\hat{F}_l$, we obtain

$$\hat{F}_l = \det[\hat{D}\Phi] \begin{bmatrix} I & 0 \\ -\delta^{-1}\hat{D}\varphi & \delta^{-1} \end{bmatrix} F_l \circ \Phi = \begin{bmatrix} \delta \hat{f}_l \\ \hat{g}_l - \hat{\operatorname{grad}}\,\varphi \cdot \hat{f}_l \end{bmatrix}$$

where $\hat{f}_l$ is the vector whose ith component is $\hat{f}_{li}(\hat{x}, \hat{t}, \hat{u})$ and $\hat{g}_l$ denotes the lth component of $\hat{g}(\hat{x}, \hat{t}, \hat{u})$. Substituting these into (3.5) and expanding, we obtain (3.3). ∎

4. Two approaches to MTP schemes. Theorem 2 maps the hyperbolic system to the cylinder which is a tensor product of a spatial domain $\Omega_{v^{(i)}}$ with a time interval $(0, 1)$. This opens up the possibility to construct tensor product discretizations – rather than spacetime discretizations – within each tent.

We denote by $\mathcal{T}_i$ the spatial mesh of $\Omega_{v^{(i)}}$ consisting of elements of $\mathcal{T}$ having $v^{(i)}$ as a vertex. For the spatial discretization, we use a finite element space X_i based on the mesh $\mathcal{T}_i$. In order to discretize (3.3), we multiply it with a spatial test function

v in X_i, integrate over the vertex patch $\Omega_{v^{(i)}}$, and manipulate the terms to get an equation of the form

$$\int_{\Omega_{v^{(i)}}} \hat{\partial}_t \hat{U}(\hat{x},\hat{t}) \cdot v(\hat{x})\, d\hat{x} = S_i(\hat{t},\hat{u},v), \tag{4.1}$$

for all $\hat{t} \in (0,1)$ and $v \in X_i$. Details of the spatial discretization, yet unspecified, are lumped into S_i. Note that the temporal derivative occurs only in the first term and can be discretized using Runge-Kutta or other schemes. Emphasizing the point that spatial discretization is thus separated from temporal discretization, we continue, leaving time undiscretized, to discuss two semidiscrete approaches. Note that both approaches proceed under the assumption that the causality condition (2.3) holds.

4.1. First approach. Recalling that $\hat{U}$ depends on $\hat{u}$, the first approach discretizes $\hat{u}(\cdot,\hat{t})$ in X_i. Let the functions $\psi_n : \Omega_{v^{(i)}} \to \mathbb{R}^L$, for $n = 1,\dots,P$, form a basis of X_i. We seek an approximation to $\hat{u}$ of the form $\hat{u}_h(\hat{x},\hat{t}) = \sum_{n=1}^{P} \mathtt{u}_n(\hat{t})\,\psi_n(\hat{x})$ where $\mathtt{u}(\hat{t})$, the vector whose nth entry is $\mathtt{u}_n(\hat{t})$, is to be found. Substituting this into (4.1) and using (3.2), we obtain $\int_{\Omega_{v^{(i)}}} \hat{\partial}_t \hat{G}(\hat{u}_h)\cdot v\, d\hat{x} = S_i(\hat{t},\hat{u}_h,v)$, for all $v \in X_i$ and $\hat{t}$ in $(0,1)$. To view this as a finite-dimensional system of ordinary differential equations (ODEs), define two maps $\mathtt{G}$ and $\mathtt{S}$ on $\mathbb{R}^P$ by

$$[\mathtt{G}(\mathtt{w})]_m = \int_{\Omega_{v^{(i)}}} \hat{G}\left(\sum_{n=1}^{P} \mathtt{w}_n \psi_n(\hat{x})\right)\psi_m(\hat{x})\, d\hat{x}, \quad [\mathtt{S}(\mathtt{w})]_m = S_i\left(\hat{t}, \sum_{n=1}^{P} \mathtt{w}_n\psi_n,\ \psi_m\right).$$

Then, putting $v = \psi_n$, we obtain the semidiscrete problem of finding a $\mathtt{u} : (0,1) \to \mathbb{R}^P$, given initial values $\mathtt{u}(0)$, satisfying the ODE system

$$\frac{d}{d\hat{t}}\mathtt{G}(\mathtt{u}(\hat{t})) = \mathtt{S}(\mathtt{u}(\hat{t})), \qquad 0 < \hat{t} < 1. \tag{4.2}$$

4.2. Second approach. The second approach discretizes $\hat{U}$ rather than $\hat{u}$, assuming that $\hat{G}^{-1}$ is at hand. We substitute $\hat{u} = \hat{G}^{-1}(\hat{U})$ into the right hand side of (4.1) and obtain the following semidiscrete problem. Find $\hat{U}_h$ of the form

$$\hat{U}_h(\hat{x},\hat{t}) = \sum_{n=1}^{P} \mathtt{U}_n(\hat{t})\psi_n(\hat{x}) \tag{4.3}$$

that satisfies $\int_{\Omega_{v^{(i)}}} \hat{\partial}_t \hat{U}_h \cdot v\, d\hat{x} = S_i(\hat{t},\hat{G}^{-1}(\hat{U}_h),v)$, for all $v \in X_i$ and $\hat{t}$ in $(0,1)$. With

$$\mathtt{M}_{mn} = \int_{\Omega_{v^{(i)}}} \psi_n(\hat{x})\psi_m(\hat{x})\, d\hat{x}, \quad [\mathtt{R}(\mathtt{w})]_m = S_i\left(\hat{t}, \hat{G}^{-1}\left(\sum_{n=1}^{P} \mathtt{w}_n\psi_n\right), \psi_m\right).$$

we obtain the following ODE system for $\mathtt{U}$, the vector of coefficients $\mathtt{U}_n(t)$.

$$\frac{d}{d\hat{t}}\mathtt{M}\mathtt{U}(\hat{t}) = \mathtt{R}(\mathtt{U}(\hat{t})), \qquad 0 < \hat{t} < 1. \tag{4.4}$$

Comparing with (4.2), instead of a possibly nonlinear $\mathtt{G}$, we now have a linear action of the mass matrix $\mathtt{M}$ in $\mathbb{R}^{P\times P}$.

4.3. Examples. We first illustrate how to treat a very general linear hyperbolic system using the first approach. In the second example we illustrate the second approach using a simple nonlinear conservation law.

Example 3 (Linear hyperbolic systems). Suppose that $A^{(j)} : \Omega_0 \to \mathbb{R}^{L\times L}$, for $j = 1, \dots, N$, are symmetric matrix-valued functions and $B : \Omega \to \mathbb{R}^{L\times L}$ is bounded. In addition, suppose $A^{(t)} \equiv A^{(N+1)}$ is a symmetric positive definite matrix-valued function from Ω_0 to $\mathbb{R}^{L\times L}$. A large class of linear examples can be obtained by setting

$$[f(x,t,u)]_{lj} = \sum_{m=1}^{L} A^{(j)}_{lm}(x)u_m, \qquad [g(x,t,u)]_l = \sum_{m=1}^{L} A^{(t)}_{lm}(x)u_m. \tag{4.5}$$

and $b(x,t,u) = B(x,t)u$. Then (1.1) can be written as

$$\partial_t(A^{(t)}u) + \sum_{j=1}^{N} \partial_j(A^{(j)}u) + Bu = 0. \tag{4.6}$$

A simple equation that fits into this example is the scalar *transport equation*. The transport of a scalar density u along a given divergence-free vector field $\beta : \Omega_0 \to \mathbb{R}^N$ is described by $\partial_t u + \operatorname{div}(\beta u) = 0$. This fits in the setting of (4.6) with $L = 1$, $B = 0$, $A^{(t)} = [1]$, and $A^{(j)}(x) = [\beta_j(x)]$, for $j = 1, 2, \dots, N$. A more complex system that also fits into this example is *electromagnetic wave propagation*. Given positive functions ε, μ and σ on Ω_0, the Maxwell system for electric field E and magnetic field H consists of $\varepsilon\partial_t E - \operatorname{curl} H + \sigma E = 0$ and $\mu\partial_t H + \operatorname{curl} E = 0$. This system also fits into (4.6) with $N = 3$, $L = 6$, and $u = \left[\begin{smallmatrix} E \\ H \end{smallmatrix}\right]$ and

$$A^{(j)} = \begin{bmatrix} 0 & [\epsilon^j] \\ [\epsilon^j]^t & 0 \end{bmatrix}, \quad A^{(t)} = \begin{bmatrix} \varepsilon I & 0 \\ 0 & \mu I \end{bmatrix}, \quad B = \begin{bmatrix} \sigma & 0 \\ 0 & 0 \end{bmatrix}$$

where ϵ^j is the matrix whose (l,m)th entry is the alternator ϵ_{jlm}.

To solve (4.6) for u on a spacetime tent K_i, we first map (4.6) to the spacetime cylinder $\hat{K}_i$ using Theorem 2. We find that the map $\hat{u} \to \hat{U}$ is now given by $\hat{U} = \hat{G}(\hat{u}) = H(\hat{x},\hat{t})\hat{u}$ where $H : \hat{K}_i \to \mathbb{R}^{L\times L}$ is the matrix function

$$H = A^{(t)} - \sum_{j=1}^{N} \Big[(1-\hat{t})\hat{\partial}_j\tau_{i-1} + \hat{t}\hat{\partial}_j\tau_i\Big]A^{(j)}. \tag{4.7}$$

Following the first approach, we discretize the term $\hat{\partial}_t(H\hat{u})$ in that form. The semidiscretization (4.2) now takes the form

$$\frac{d}{d\hat{t}}\left(\mathtt{H}(\hat{t})\mathtt{u}(\hat{t})\right) = \mathtt{S}(\mathtt{u}(\hat{t})), \qquad 0 < \hat{t} < 1, \tag{4.8}$$

where $\mathtt{H}$ is the matrix whose entries are $\mathtt{H}_{mn}(\hat{t}) = \displaystyle\int_{\Omega_{v(i)}} H(\hat{x},\hat{t})\,\psi_n(\hat{x})\cdot\psi_m(\hat{x})\,d\hat{x}$.

Example 4 (2D inviscid scalar Burgers equation). A simple two-dimensional analogue of the well-known one-dimensional inviscid Burgers equation is the following scalar conservation law considered in [11]. In the framework leading to (1.1), set $L = 1$, $N = 2$, $g(x,t,u) = u$, $f(x,t,u) = \frac{1}{2}u^2\begin{bmatrix}1 & 1\end{bmatrix}$, and $b \equiv 0$ to get $\partial_t u + \frac{1}{2}\left(\partial_1(u^2) + \partial_2(u^2)\right) = 0$. Applying Theorem 2 to map this equation from a tent K_i to the spacetime cylinder $\hat{K}_i$, we find that $\hat{U} = \hat{G}(\hat{u})$ satisfies

$$\hat{U} = \hat{u} - \frac{1}{2}\hat{u}^2 d \tag{4.9}$$

where $d = \hat{\partial}_1\varphi + \hat{\partial}_2\varphi$. To illustrate how to use the second approach, we compute $\hat{u} = \hat{G}^{-1}(\hat{U})$ by solving the quadratic equation $d\hat{u}^2 - 2\hat{u} + 2\hat{U} = 0$. The roots are

$$\hat{u} = \frac{1 \pm \sqrt{1 - 2d\hat{U}}}{d} = \frac{2\hat{U}}{1 \mp \sqrt{1 - 2d\hat{U}}}. \tag{4.10}$$

We will now show that the causality condition (2.3) implies that these roots are real and only one of the two roots is valid. Recall that the maximal characteristic speed $\bar{c}$ is given by the largest eigenvalue of $D_u f \cdot \nu$ for all unit vectors $\nu \in \mathbb{R}^2$. For this example the maximum is achieved at $\nu = \pm\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\end{bmatrix}$ and thus $\bar{c}$ is given by $\bar{c} = |D_u f \cdot \nu| = \sqrt{2}|\hat{u}|$. Thus, the causality condition (2.3) yields

$$|\mathrm{grad}_x\varphi| = |(1-\hat{t})\mathrm{grad}_x\tau_{i-1} + \hat{t}\,\mathrm{grad}_x\tau_i| < \frac{1}{\bar{c}} = \frac{1}{\sqrt{2}|\hat{u}|}.$$

Since $|d| \le |\hat{\partial}_1\varphi| + |\hat{\partial}_2\varphi| \le \sqrt{2(\hat{\partial}_1\varphi)^2 + 2(\hat{\partial}_2\varphi)^2}$, this implies

$$|\hat{u}d| < 1. \tag{4.11}$$

Rewriting (4.9) as $2\hat{U}d = \hat{u}d(2 - \hat{u}d)$, we see that (4.11) implies $1 - 2d\hat{U} \ge 0$, so the roots in (4.10) exist.

Finally, observing that the first equality of (4.10) can be written as

$$\hat{u}d - 1 = \pm\sqrt{1 - 2d\hat{U}}, \tag{4.12}$$

we conclude from (4.11) that $\hat{u}d - 1 < |\hat{u}d| - 1 < 0$, i.e., we must choose the negative sign in the $\pm$ on the right hand side of (4.12). Thus we obtain the correct root

$$\hat{G}^{-1}(\hat{U}) = \frac{2\hat{U}}{1 + \sqrt{1 - 2d\hat{U}}}.$$

One can now proceed with the second approach by applying a standard spatial discontinuous Galerkin discretization and time stepping by a Runge-Kutta scheme. Some regularization or slope limiting technique is needed to avoid spurious oscillations near sharp solution transitions. This issue is considered further in Section 6.

5. A locally implicit MTP scheme for the wave equation.

5.1. The acoustic wave problem. Suppose we are given a material coefficient $\alpha : \Omega_0 \to \mathbb{R}^{N\times N}$, symmetric and positive definite everywhere in Ω_0 and a damping coefficient $\beta : \Omega_0 \to \mathbb{R}$. The wave equation for the linearized pressure $\phi : \Omega \to \mathbb{R}$ is

$$\partial_{tt}\phi + \beta\partial_t\phi - \mathrm{div}_x(\alpha \mathrm{grad}_x\phi) = 0 \qquad \text{in } \Omega. \tag{5.1a}$$

While a variety of initial and boundary conditions are admissible in MTP schemes, for definiteness, we focus on these model conditions:

$$n_x \cdot \alpha \mathrm{grad}_x\phi = 0 \qquad \text{on } \partial\Omega_0 \times (0, t_{\max}), \tag{5.1b}$$

$$\partial_t\phi = \phi_1 \text{ and } \phi = \phi_0 \qquad \text{on } \Omega_0 \times \{0\}. \tag{5.1c}$$

for some given sufficiently smooth compatible data ϕ_0 and ϕ_1. In (5.1b), n_x denotes the spatial component of the outward unit normal.

Let us put (5.1) into the framework of (1.1) using Example 3. Set $L = N+1$ and

$$u = \begin{bmatrix} q \\ \mu \end{bmatrix} = \begin{bmatrix} \alpha\mathrm{grad}_x\phi \\ \partial_t\phi \end{bmatrix} \in \mathbb{R}^L.$$

Then (5.1a) yields $\alpha^{-1}\partial_t q - \mathrm{grad}_x\mu = 0$ and $\partial_t\mu - \mathrm{div}\, q - \beta\mu = 0$. This is readily identified to be in the form (4.6) with

$$A^{(t)} = \begin{bmatrix} \alpha^{-1} & 0 \\ 0 & 1 \end{bmatrix}, \quad A^{(j)} = -\begin{bmatrix} 0 & e_j \\ e_j^t & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & \beta \end{bmatrix},$$

where e_j denotes the jth unit (column) vector. The boundary condition in the new variable is $n_x \cdot q = 0$ on $\partial\Omega_0 \times (0, t_{\max})$, and the initial conditions take the form $q = \alpha\mathrm{grad}_x\phi_0$ and $\mu = \phi_1$ on Ω_0.

To describe the MTP scheme, set $u_0 = (q_0, \mu_0) = (\alpha\mathrm{grad}_x\phi_0, \phi_1)$. Suppose we are at the ith tent pitching step. Then the solution $u_{i-1} = (q_{i-1}, \mu_{i-1})$ has been computed on the advancing front S_{i-1}, and a new tent K_i has been erected at mesh vertex $v^{(i)}$. We now need the wave equation mapped over to $\hat{K}_i = \Omega_{v^{(i)}} \times (0, 1)$. From Example 3,

$$\frac{\partial}{\partial\hat{t}}(H\hat{u}) + \sum_{j=1}^{N} \frac{\partial}{\partial\hat{x}_j}(\delta A^{(j)}\hat{u}) + \delta\hat{B}\hat{u} = 0, \tag{5.2}$$

where H is as in (4.7) and $\hat{B} = B \circ \Phi$ has the sole nonzero entry $\hat{\beta} = \beta \circ \Phi$. In this example, it is convenient to split $\hat{u}$ into two blocks consisting of $\hat{q} = q \circ \Phi \in \mathbb{R}^N$ and $\hat{\mu} = \mu \circ \Phi \in \mathbb{R}$. Then for all $(\hat{x}, \hat{t}) \in \hat{K}_i$,

$$H(\hat{x}, \hat{t}) \begin{bmatrix} \hat{q} \\ \hat{\mu} \end{bmatrix} = \begin{bmatrix} \hat{\alpha}^{-1} & \mathrm{grad}_x\varphi \\ (\mathrm{grad}_x\varphi)^t & 1 \end{bmatrix} \begin{bmatrix} \hat{q} \\ \hat{\mu} \end{bmatrix} \tag{5.3}$$

where $\hat{\alpha} = \alpha \circ \Phi$ and (5.2) can be rewritten as

$$\frac{\partial}{\partial\hat{t}} \begin{bmatrix} \hat{\alpha}^{-1}\hat{q} + \hat{\mu}\mathrm{grad}_x\varphi \\ \hat{\mu} + \hat{q} \cdot \mathrm{grad}_x\varphi \end{bmatrix} - \begin{bmatrix} \mathrm{grad}_x(\delta\hat{\mu}) \\ \mathrm{div}(\delta\hat{q}) \end{bmatrix} + \begin{bmatrix} 0 \\ \delta\hat{\beta}\hat{\mu} \end{bmatrix} = 0 \qquad \text{in } \Omega_{v^{(i)}} \times (0, 1). \tag{5.4}$$

On the cylinder, this equation must be supplemented by the initial conditions $\hat{q} = \hat{q}_{i-1}$ and $\hat{\mu} = \hat{\mu}_{i-1}$ on $\Omega_{v^{(i)}} \times \{0\}$.

5.2. Semidiscretization after mapping. For the spatial discretization, we use the Brezzi-Douglas-Marini (BDM) mixed method. Namely, letting $P_p(T)$ denote the space of polynomials of degree at most p in $\hat{x}$, restricted to a spatial N-simplex T, set $X_i = \{(r,\eta) \in H(\mathrm{div}, \Omega_{v^{(i)}}) \times L^2(\Omega_{v^{(i)}}) : r|_T \in P_p(T)^N$ and $\eta|_T \in P_p(T)$ for all simplices $T \in \mathcal{T}_i$ and $r \cdot n_x = 0$ on $\partial\Omega_{v^{(i)}} \cap \partial\Omega_0\}$. Multiplying (5.4) by (r,η) and integrating the first equation by parts, we obtain

$$\frac{d}{d\hat{t}} \int_{\Omega_{v^{(i)}}} \begin{bmatrix} \hat{\alpha}^{-1}\hat{q} + \hat{\mu}\mathrm{grad}_x\varphi \\ \hat{\mu} + \hat{q}\cdot\mathrm{grad}_x\varphi \end{bmatrix} \cdot \begin{bmatrix} r \\ \eta \end{bmatrix} d\hat{x} = \int_{\Omega_{v^{(i)}}} \begin{bmatrix} -\delta\hat{\mu} \\ \mathrm{div}(\delta\hat{q}) - \delta\hat{\beta}\hat{\mu} \end{bmatrix} \cdot \begin{bmatrix} \mathrm{div}\, r \\ \eta \end{bmatrix} d\hat{x}, \tag{5.5}$$

for all $(r,\eta) \in X_i$. Using a basis $\psi_m \equiv (r_m, \eta_m)$ of X_i, the coefficients $\mathtt{u}_m(\hat{t})$ of the expansion of $\hat{u}$ in this basis satisfy an ODE system, which can be written using matrices $\mathtt{H}$ and $\mathtt{S}$ defined by

$$\mathtt{H}_{lm}(\hat{t}) = \int_{\Omega_{v^{(i)}}} \begin{bmatrix} \hat{\alpha}^{-1} r_m + \eta_m \mathrm{grad}_x\varphi \\ \eta_m + r_m \cdot \mathrm{grad}_x\varphi \end{bmatrix} \cdot \begin{bmatrix} r_l \\ \eta_l \end{bmatrix} d\hat{x} \tag{5.6a}$$

$$\mathtt{S}_{lm} = \int_{\Omega_{v^{(i)}}} \begin{bmatrix} -\delta\eta_m \\ \mathrm{div}(\delta r_m) - \delta\hat{\beta}\eta_m \end{bmatrix} \cdot \begin{bmatrix} \mathrm{div}\, r_l \\ \eta_l \end{bmatrix} d\hat{x}. \tag{5.6b}$$

Using prime ($'$) to abbreviate $d/d\hat{t}$, observe that (5.5) is the same as

$$\big(\mathtt{H}(\hat{t})\mathtt{u}(\hat{t})\big)' = \mathtt{S}\mathtt{u}(\hat{t}), \qquad 0 < \hat{t} < 1, \tag{5.6c}$$

a realization of (4.8) for the wave equation.

5.3. Time discretization after mapping. We utilize the first approach of §4.1. by applying an implicit high order multi-stage Runge-Kutta (RK) method of Radau IIA type [10, Chapter IV.5] for time stepping (5.6c). Note that due to the implicit nature of the scheme, there is no CFL constraint on the number of stages (within the mapped tent), irrespective of the spatial polynomial degree p of X_i. These RK methods, with s stages, are characterized by numbers a_{lm} and c_l for $l, m = 1, \ldots, s$ (forming entries of a Butcher tableau) with the property that $c_s = 1$ (and the remaining c_l are determined by the roots of appropriate Jacobi polynomials). When applied to a standard ODE $y' = f(\hat{t}, y)$ in the interval $\hat{t} \in (0,1)$, with initial condition $y(0) = y_0$, it produces approximations y_l to y at $t_l = c_l$ that satisfy

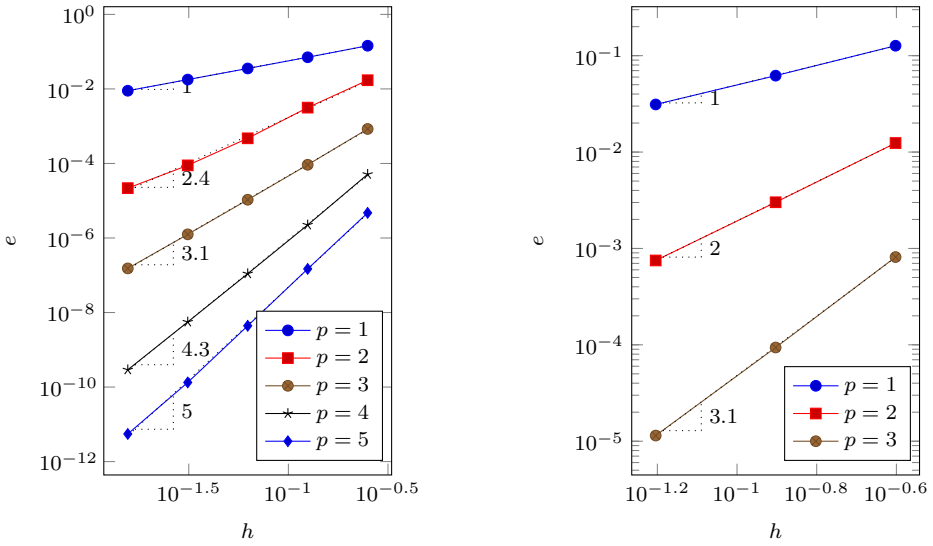
$$y_l = y_0 + \sum_{m=1}^{s} a_{lm} f(t_m, y_m), \qquad l = 1, \ldots, s. \tag{5.7}$$

However, since (5.6c) is not in this standard form, we substitute $y_l = \mathtt{H}_l\mathtt{u}_l$ into (5.7), where $\mathtt{H}_l = \mathtt{H}(\hat{t}_l)$ and $\mathtt{u}_l$ is the approximation to $\mathtt{u}(t_l)$ to be found. Also setting $f(t_m, y_m) = \mathtt{S}\mathtt{u}_m$, we obtain the linear system

$$\mathtt{H}_l\mathtt{u}_l = \mathtt{H}_0\mathtt{u}_0 + \sum_{m=1}^{s} a_{lm}\mathtt{S}\mathtt{u}_m, \qquad l = 1, \ldots, s,$$

which can be easily solved for the final stage solution $\mathtt{u}_s$, given $\mathtt{u}_0$.

5.4. Numerical studies in two and three space dimensions. The locally implicit MTP method was implemented within the framework of the NGSolve [20] package. We report the results obtained for (5.1) with $\beta = 0$, $\alpha = 1$, Ω_0 set to the



(a) Example in two space dimensions

(b) Example in three space dimensions

Fig. 3: Convergence rates for a standing wave

unit square, $\phi_0 = 0$ and $\phi_1 = \cos(\pi x_1)\cos(\pi x_2)$ for $(x_1, x_2) \in \Omega_0$. It is easy to see that the exact solution is the classical standing wave

$$\phi(x,t) = \cos(\pi x_1)\cos(\pi x_2)\sin(\pi t\sqrt{2})/(\sqrt{2}\pi),$$

and

$$u(x,t) = \begin{bmatrix} q(x,t) \\ \mu(x,t) \end{bmatrix} = \begin{bmatrix} \text{grad}_x \phi \\ \partial_t \phi \end{bmatrix} = \begin{bmatrix} -\sin(\pi x_1)\cos(\pi x_2)\sin(\pi t\sqrt{2})/\sqrt{2} \\ -\cos(\pi x_1)\sin(\pi x_2)\sin(\pi t\sqrt{2})/\sqrt{2} \\ \cos(\pi x_1)\cos(\pi x_2)\cos(\pi t\sqrt{2}) \end{bmatrix}.$$

The spatial domain Ω_0 is meshed by a uniform grid obtained by dividing the unit square into $2^l \times 2^l$ congruent squares and dividing each square into two triangles by connecting its positively sloped diagonal. The parameters to be varied in each experiment are the spatial mesh size $h = 2^{-l}$ and the the polynomial degree p of the space discretization. The number of Runge-Kutta time stages is fixed to $s = p$. The tent meshing algorithm is driven by an input wavespeed of 2 (leading to conservative tent pole heights) to mesh a time slab of size $2^{-l}/8$. This time slab is stacked in time to mesh the entire spacetime region of simulation $\Omega_0 \times (0, 1)$. Letting $q_h(x)$ and $\mu_h(x)$ denote the computed solutions at time $t = 1$, we measure the error norm e defined by $e^2 = \|q(\cdot, 1) - q_h\|^2_{L^2(\Omega_0)} + \|\mu(\cdot, 1) - \mu_h\|^2_{L^2(\Omega_0)}$. The observations are compiled in Figure 3a, where the values of e as a function of degree p and h are plotted. The rate r of the $O(h^r)$-convergence observed is computed from the slope of the regression lines and marked near each convergence curve. We observe that e appears to go to 0 at a rate of $O(h^p)$.

Next, consider the case of three spatial dimensions, where Ω_0 is set to the unit cube and subdivided in a fashion analogous to the two-dimensional case (into $2^l \times 2^l \times 2^l$

congruent cubes, which are further subdivided into six tetrahedra). The remaining parameters are the same as in the two-dimensional case, except that now the exact solution is $\phi(x,t) = \cos(\pi x_1)\cos(\pi x_2)\cos(\pi x_3)\sin(\pi t\sqrt{3})/(\sqrt{3}\pi)$. Note that the spacetime mesh of tents, now formed by *four-dimensional* simplices, continues to be made by Algorithm 2. The convergence history plotted in Figure 3b shows that e, just as in the previous case, goes to zero at a rate of $O(h^p)$.

6. An explicit MTP scheme for a nonlinear conservation law. In this section, we describe some techniques for handling nonlinear conservation laws, and considering the specific example of Euler equations, construct a explicit MTP scheme.

6.1. Mapping an entropy pair. Recall that a real function $\mathcal{E}(u)$ is called an entropy [21, Definition 3.4.1] of the system (1.1) if there exists an entropy flux $\mathcal{F}(u) \in \mathbb{R}^N$ such that every classical solution u of (1.1) satisfies $\partial_t \mathcal{E}(u) + \mathrm{div}_x \mathcal{F}(u) = 0$. Note that for nonsmooth u, this equality need not hold. The pair $(\mathcal{E}, \mathcal{F})$ is called the entropy pair. We say that this pair satisfies the "entropy admissibility condition" on Ω if

$$\partial_t \mathcal{E}(u(x,t)) + \mathrm{div}_x \mathcal{F}(u(x,t)) \leq 0 \tag{6.1}$$

holds in the sense of distributions on Ω. The inequality is useful to study the violation of entropy conservation for nonsmooth solutions (like shocks). Nonlinear conservation laws often have multiple weak solutions and uniqueness is obtained by selecting a solution u satisfying the entropy admissibility condition. These theoretical considerations motivate the use of numerical analogues of (6.1) in designing schemes for conservation laws.

Suppose that on a tent K_i, we are given a solution $u(x,t)$ of (1.1) and an entropy pair $(\mathcal{E}, \mathcal{F})$. The mapped solution, as before, is $\hat{u} = u \circ \Phi$. Define

$$\hat{\mathcal{E}}(w) = \mathcal{E}(w) - \mathcal{F}(w)\mathrm{grad}_x \varphi, \qquad \hat{\mathcal{F}}(w) = \delta \mathcal{F}(w). \tag{6.2}$$

THEOREM 5. *Suppose u solves (1.1) on K_i and $\hat{u} = u \circ \Phi$ solves the mapped equation (3.3). Then, whenever $(\mathcal{E}, \mathcal{F})$ is an entropy pair for (1.1), $(\hat{\mathcal{E}}, \hat{\mathcal{F}})$ is an entropy pair for (3.3). Moreover, if $\mathcal{E}(u)$ and $\mathcal{F}(u)$ satisfies the entropy admissibility condition (6.1) on K_i, then $\hat{\mathcal{E}}(\hat{u})$ and $\hat{\mathcal{F}}(\hat{u})$ satisfies the entropy admissibility condition on $\hat{K}_i$.*

Proof. Repeating the calculations in the proof of Theorem 2, with $g = \mathcal{E}$ and $f = \mathcal{F}$, we obtain

$$(\partial_t \mathcal{E}(u) + \mathrm{div}_x \mathcal{F}(u)) \circ \Phi = \frac{1}{\delta}\left(\hat{\partial}_t \hat{\mathcal{E}}(\hat{u}) + \hat{\mathrm{div}}_x \hat{\mathcal{F}}(\hat{u})\right),$$

from which the statements of the theorem follow. ∎

6.2. Entropy viscosity regularization. The addition of "artificial viscosity" (a diffusion term) to the right hand side of nonlinear conservation laws makes their solutions dissipative. When the limit of such solutions, as the diffusion term goes to zero, exist in some sense, it is referred to as a vanishing viscosity solution. It is known [2, Theorem 4.6.1] that the vanishing viscosity solution satisfies the entropy admissibility condition for entropy pairs satisfying certain conditions. Motivated by such connections, the entropy viscosity regularization method of [9], suggests modifying numerical schemes by selectively adding small amounts of artificial viscosity, to avoid spurious oscillations near discontinuous solutions. We borrow this technique

and incorporate it into the MTP schemes obtained using the second approach (of §4.2) as follows.

Consider the problem on the tent K_i mapped to $\hat{K}_i$. We set the spatial discretization space to $X_i = \{u \in L^2(\Omega_{v^{(i)}})^L : u|_T \in P_p(T) \text{ for all } T \in \mathcal{T}_i\}$ and consider a DG discretization of the mapped equation (3.3) following the second approach. Accordingly the approximation $\hat{U}_h(\hat{x}, \hat{t})$ takes the form in (4.3). Let $(\cdot,\cdot)_h$ and $\langle\cdot,\cdot\rangle_h$ denote the sum of integrals over T and ∂T of the appropriate inner product of its arguments, over all $T \in \mathcal{T}_i$, respectively. The semidiscretization of (3.3) by the DG method takes the form

$$(\hat{\partial}_t \hat{U}_h, V)_h - (\delta\, f(\hat{G}^{-1}(\hat{U}_h)), \mathrm{grad}_x V)_h + \langle \delta Q_f(\hat{U}_h), V\rangle_h + (\delta\, b, V)_h = 0 \tag{6.3}$$

for all $V \in X_i$. Here Q_f is the so-called "numerical flux," whose form varies depending on the DG method, and as usual, all derivatives are taken element by element.

Suppose that an entropy pair $(\mathcal{E}, \mathcal{F})$ is given for (1.1). On the mapped tent $\hat{K}_i$, let $(\hat{\mathcal{E}}, \hat{\mathcal{F}})$ be defined by (6.2). Suppose a numerical approximation $\hat{U}_h(\hat{x}, \hat{t}_1)$ has been computed at some time $0 \le \hat{t}_1 < 1$ and we want to compute a numerical approximation at the next time stage, say at $\hat{t} = \hat{t}_1 + \Delta t \le 1$. The *entropy residual* of the approximation $u_h = \hat{G}^{-1}(\hat{U}_h)$ to u is a weak form of the quantity $\hat{\partial}_t \hat{\mathcal{E}}(\hat{u}_h) + \hat{\mathrm{div}}_x \hat{\mathcal{F}}(\hat{u}_h)$, which by Theorem 5, is non-positive. The discrete entropy residual at time $\hat{t}_1$ is $R_h = \min(r_h, 0)$ where $r_h \in X_i$ is defined by

$$\begin{aligned}(\delta r_h, V)_h &= (\hat{\partial}_t \hat{\mathcal{E}}(\hat{G}^{-1}(\hat{U}_h)), V)_h - (\hat{\mathcal{F}}(\hat{U}_h), \mathrm{grad}_x V)_h + \langle \delta Q_{\mathcal{F}}(\hat{U}_h), V\rangle_h \\ &= (\frac{\partial(\hat{\mathcal{E}} \circ \hat{G}^{-1})}{\partial U} \hat{\partial}_t \hat{U}_h, V)_h - (\hat{\mathcal{F}}(\hat{U}_h), \mathrm{grad}_x V)_h + \langle \delta Q_{\mathcal{F}}(\hat{U}_h), V\rangle_h\end{aligned}$$

for all $V \in X_i$. Here $Q_{\mathcal{F}}$ is a numerical flux prescribed by a DG approximation to the entropy conservation equation. The term $\hat{\partial}_t \hat{U}_h$ can be replaced by its approximation available from (6.3) while computing r_h.

Next, following [9], we quantify the amount of viscosity to be added to (6.3). Define the *entropy viscosity coefficient* on one spatial element $T \in \mathcal{T}_i$ by $\nu_e^T = c_X^2 \|R_h\|_{L^\infty(T)} / |\bar{\mathcal{E}}|$ where $\bar{\mathcal{E}}$ is the mean value of $\hat{\mathcal{E}}(\hat{G}^{-1}(\hat{U}_h))$ on T and c_X is an effective local grid size of X_i, typically chosen as $c_X = \kappa_1 \,\mathrm{diam}(T)/p$ for some fixed number κ_1. To limit the viscosity added based on local wavespeed, define $\nu_*^T = \kappa_2 \,\mathrm{diam}(T) \|D_u \hat{f}(\hat{x}, \hat{t}_1, \hat{u}_h(\hat{x}, \hat{t}_1))\|_{L^\infty(T)}$ where κ_2 is another fixed number and set $\nu_i = \max_{T \in \mathcal{T}_i} \min(\nu_*^T, \nu_e^T)$. This artificial viscosity coefficient proposed in [9] leads to generous viscosity at discontinuities (where the entropy residual is high) and little viscosity in smooth regions. Finally, we modify the mapped equation (3.3) by adding to its right hand side the corresponding artificial viscosity term $\nu_i \hat{\mathrm{div}}_x(\delta \mathrm{grad}_x \hat{u})$. Namely, instead of solving (6.3) for $\hat{t}_1 \le \hat{t} \le \hat{t}_1 + \Delta t$, we solve its viscous perturbation:

$$\begin{aligned}(\hat{\partial}_t \hat{U}_h, V)_h - (\delta\, f(\hat{G}^{-1}(\hat{U}_h)), \mathrm{grad}_x V)_h + \langle \delta Q_f(\hat{U}_h), V\rangle_h + (\delta\, b, V)_h \\ + \nu_i a_i(\hat{G}^{-1}(\hat{U}_h), V) = 0,\end{aligned} \tag{6.4}$$

for all $v \in X_i$, where $a_i(\cdot,\cdot)$ is the standard interior penalty DG approximation of the viscous term $-\hat{\mathrm{div}}_x(\delta \mathrm{grad}_x \hat{u})$ defined below. On an interface F shared by two elements T_+ and T_-, with outward unit normals n_+ and n_-, respectively, set $[wn] = w|_{T_+} n_+ + w|_{T_-} n_-$, with the understanding that $w(x,t)$ is considered zero if x is

outside $\Omega_{v^{(i)}}$. Then

$$a_i(w,v) = (\delta \mathrm{grad}_x w, \mathrm{grad}_x v)_h - \frac{1}{2}\langle \delta \mathrm{grad}_x w, [vn]\rangle_h - \frac{1}{2}\langle [wn], \delta \mathrm{grad}_x v\rangle_h + \frac{\alpha}{2h}\langle \delta[wn], [vn]\rangle_h.$$

Here, as usual, the penalization parameter α must be chosen large enough to obtain coercivity. Applying a time stepping algorithm to (6.4), we compute the numerical solution at the next time stage $t_1 + \Delta t$.

6.3. Application to Euler equations. Let $\rho : \Omega \to \mathbb{R}$, $m : \Omega \to \mathbb{R}^N$ and $E : \Omega \to \mathbb{R}$ denote the density, momentum, and total energy of a perfect gas occupying $\Omega \subset \mathbb{R}^N$. Set $L = N + 2$ and let

$$u = \begin{bmatrix} \rho \\ m \\ E \end{bmatrix}, \quad g(u) = u\,, \quad f(u) = \begin{bmatrix} m \\ \mathrm{P} I + m \otimes m/\rho \\ (E + \mathrm{P})m/\rho \end{bmatrix}, \quad b \equiv 0\,,$$

Here, the pressure P is related to the state variables by $\mathrm{P} = \frac{1}{2}\rho \mathrm{T}$, and $\mathrm{T} = \frac{4}{d}(\frac{E}{\rho} - \frac{1}{2}\frac{|m|^2}{\rho^2})$, where d, the degrees of freedom of the gas particles, is set to 5 for ideal gas. With these settings, the system of Euler equations is given by (1.1).

After mapping from a tent K_i to $\hat{K}_i$, to proceed with the second approach we need to invert the nonlinear equation $\hat{U} = \hat{G}(\hat{u})$. Namely, writing $\hat{u} = (\hat{\rho}, \hat{m}, \hat{E})$ and $\hat{U} = (\hat{R}, \hat{M}, \hat{F})$, we want to explicitly compute $(\hat{\rho}, \hat{m}, \hat{E}) = \hat{G}^{-1}(\hat{R}, \hat{M}, \hat{F})$. Lengthy calculations (see [25]) show that the expression for $\hat{G}^{-1}$ is given by

$$\hat{\rho} = \frac{\hat{R}^2}{a_1 - \frac{2}{d}|\mathrm{grad}_x\varphi|^2 a_3}, \quad \hat{m} = \frac{\hat{\rho}}{\hat{R}}(\hat{M} + \frac{2}{d}a_3 \mathrm{grad}_x\varphi), \quad \hat{E} = \frac{\hat{\rho}}{\hat{R}}(\hat{F} + \frac{2a_3}{d\hat{\rho}}\mathrm{grad}_x\varphi \cdot \hat{m})$$

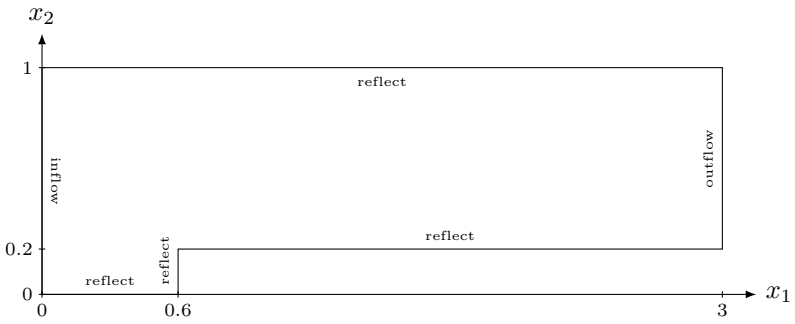
where

$$a_1 = \hat{R} - \hat{M} \cdot \mathrm{grad}_x\varphi, \qquad a_2 = 2\hat{F}\hat{R} - |\hat{M}|^2,$$
$$a_3 = a_2/(a_1 + \sqrt{a_1^2 - \frac{4(d+1)}{d^2}|\mathrm{grad}_x\varphi|^2 a_2}).$$

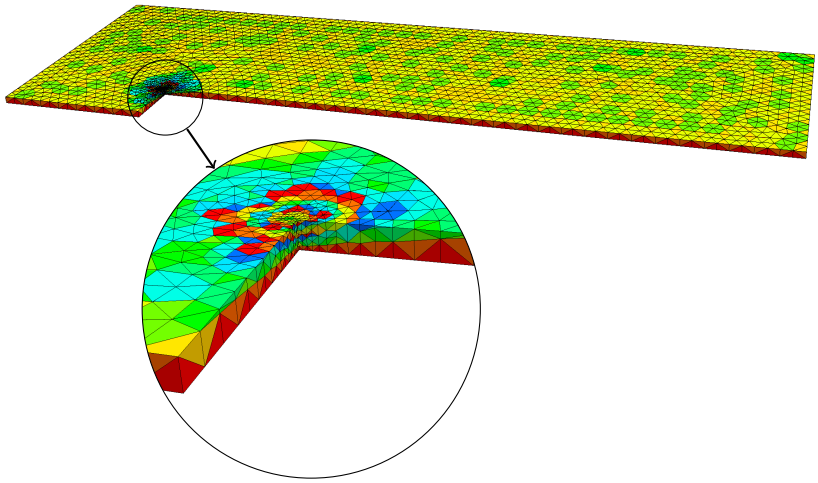
The well-known expressions for the entropy and entropy flux for the Euler system are $\mathcal{E}(\rho, m, E) = \rho\left(\ln\rho - \frac{d}{2}\ln \mathrm{T}\right)$ and $\mathcal{F}(\rho, m, E) = m\mathcal{E}/\rho$. With these expressions we discretize the mapped equation using the second approach, applying the previously described entropy viscosity regularization of (6.4).

6.4. A computational illustration. We consider the well-known example [26] of a wind tunnel with a forward facing step on which a Mach 3 flow impinges. The geometry is shown in Figure 4a and the initial conditions are set to $\rho = 1.4$, $m = \rho\begin{bmatrix}3 & 0\end{bmatrix}^t$, and $\mathrm{P} = 1$. The boundary conditions are set such that $(0, x_2)$ is an inflow boundary and $(3, x_2)$ is a free boundary, which has no effect on the flow. All other boundaries are solid walls. Anticipating the singularity at the nonconvex corner, we construct a spatial mesh with small elements near it. Figure 4b shows this mesh and the unstructured locally adaptive time advance that is possible.

Using the notation of (4.4) and a basis ψ_l of X_i, we obtain the ODE system $(\mathtt{MU}(\hat{t}))' = \mathtt{R}^1(\mathtt{U}(\hat{t})) - \mathtt{R}^2(\mathtt{U}(\hat{t}))$, for $0 < \hat{t} < 1$, where $[\mathtt{R}^1]_l = (\delta f(\hat{G}^{-1}(\hat{U}_h)), \mathrm{grad}_x\psi_l)_h - \langle \delta Q_f(\hat{U}_h), \psi_l\rangle_h$ and $[\mathtt{R}^2]_l = \nu_i a_i(\hat{G}^{-1}(\hat{U}_h), \psi_l)$. This system within each tent is solved by a time stepping scheme and a time step $\Delta t = \frac{1}{m}$, where m denotes the number of local time steps. For stability we need $m \geq O(p^2)$, but more time steps may be used for



(a) Geometry and boundary conditions



(b) Locally refined tent mesh of one time slab and zoomed in view. (The tent colors cycle through the parallel layer numbers – see Remark 1.)

accuracy. Due to the addition of artificial viscosity, an additional fractional time step Δt_v is chosen depending on the viscosity coefficient (and therefore on the smoothness of the solution). Details based on the explicit Euler method are in Algorithm 3, where we use the notations $\mathtt{U}^j := \mathtt{U}(j\Delta t)$ and $\delta_* = \|\delta\|_{L^\infty(\Omega_{v(i)})}$.

Algorithm 3 Addition of artificial viscosity

For $j = 0, \ldots, m-1$ do:

- Evaluate $\mathtt{R}^1(\mathtt{U}^j)$.
- Update solution $\mathtt{U}^{j+1} = \mathtt{U}^j + \Delta t\, \mathtt{R}^1(\mathtt{U}^j)$.
- Calculate the entropy residual and the viscosity coefficient $\nu_i(j\Delta t)$.
- Estimate time step $\Delta t_v = \Delta t / \frac{\delta_* \nu_i p^4}{h^2}$ for the artificial viscosity.
- Apply the artificial viscosity with an explicit Euler method up to the time $(j+1)\Delta t$.

This algorithm can be generalized for any Runge-Kutta scheme and for the following results a two-staged RK scheme was used. A kinetic flux (see [13]) was used

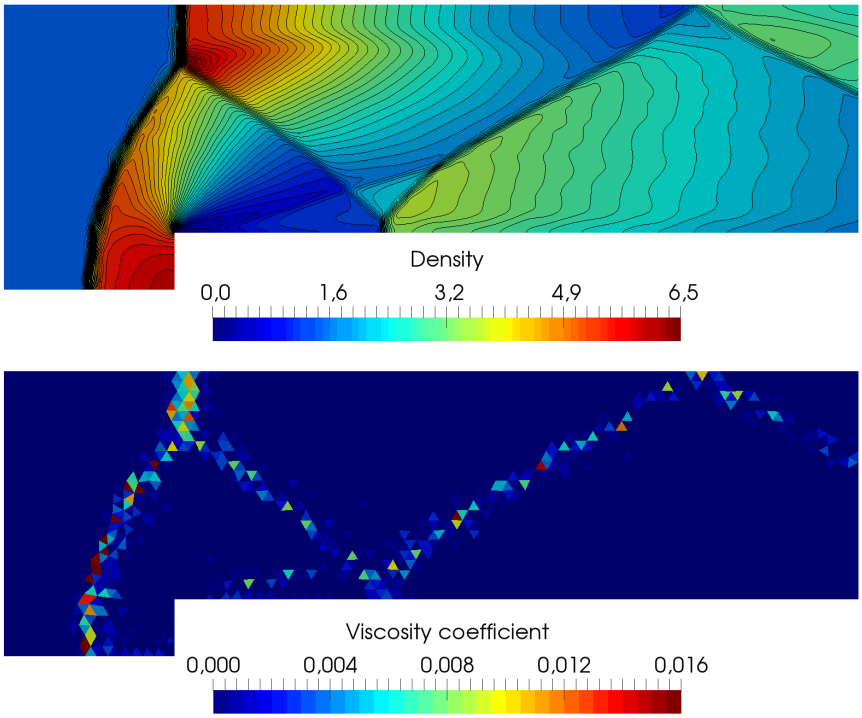


Fig. 5: Solution of Mach 3 wind tunnel at $t = 4$, with $p = 4$ DG finite elements on 3951 triangles

for the numerical flux Q_f while $Q_{\mathcal{F}}$ was set by

$$Q_{\mathcal{F}} = \begin{cases} \mathcal{F}(\hat{\rho}^+, \hat{m}^+, \hat{E}^+) \cdot n\,, & \hat{m}^+ \cdot n \geq 0\,, \\ \mathcal{F}(\hat{\rho}^-, \hat{m}^-, \hat{E}^-) \cdot n\,, & \text{otherwise}\,, \end{cases}$$

where $\hat{\rho}^+$ denotes the trace of $\hat{\rho}$ from within the element which has n as outward unit normal vector. For computational convenience, we use a slight variation of the entropy viscosity regularization described in §6.2. Namely, the entropy viscosity coefficient on one element $T \in \mathcal{T}_i$ is set by $\nu_e^T = c_X^2 \|R_h\|_{L^\infty(T)}$ and the limiting artificial viscosity is set by $\nu_*^T = \kappa_2 \operatorname{diam}(T) \|\rho(|\frac{m}{\rho}| + \sqrt{\gamma T})\|_{L^\infty(T)}$ with $\gamma = \frac{d+2}{d} = 1.4$ for an ideal gas and the temperature T. The constants in the calculation of the entropy viscosity coefficient were chosen as $\kappa_1 = \frac{1}{2}$, $\kappa_2 = \frac{1}{4p}$ and the penalization parameter α in the artificial viscosity term is set to 2.

With these settings, the results obtained with $p = 4$, are shown in Figure 5. They correspond to the results [26] that can be found in the literature using other methods. Note from the second plot that the artificial viscosity is applied only in the shocks.

7. Conclusion. We have introduced new schemes, called MTP schemes, for advancing hyperbolic solutions through unstructured tent meshes. The advantages of tent pitching over traditional time stepping, amply clarified by others in the literature on SDG methods, include the ability to advance in time by different amounts at different spatial locations, easy parallelization, and linear scaling of computational complexity in the number of tents.

Further new advantages brought about by MTP schemes include the possibility to use existing spatial discretizations and time stepping schemes after mapping tents to cylinders. The mapping technique has opened a new avenue to perform fully explicit matrix-free local time stepping on unstructured tent meshes using explicit MTP schemes. Their utility as a powerful computational tool was demonstrated on the Mach 3 wind tunnel where local refinement near a rarefaction singularity permitted us to capture the shock structure using relatively few elements and standard discretizations (in separated space and time).

We also studied locally implicit MTP schemes and their application to the acoustic wave equation. We observed $O(h^p)$ accuracy in the L^2 norm on the final time slice for smooth solutions. To compare with SDG schemes, to our knowledge, the only known provable error estimate is $O(h^{p+1/2})$ [15], although $O(h^{p+1})$ is often observed in practice [1, 15]. Further theoretical studies are needed to understand definitively the cause of this loss of convergence order and whether no scheme can expect faster convergence than $O(h^{p+1/2})$ on all meshes. Nonetheless, for the moment, proceeding by handicapping MTP schemes by one full order (and requiring MTP schemes to use one higher p), let us see what might be gained by the separation of space and time in the MTP scheme: As p increases, SDG schemes use $O(p^{N+1})$ spacetime basis functions per tent, while MTP schemes use $O((p+1)^N)$ spatial basis functions to obtain the same convergence rate. Hence to propagate the solution inside a tent, an SDG scheme performs $O(p^{2(N+1)})$ flops, while the implicit MTP scheme performs $O((p+1)^{2N})$ flops. Since $(p+1)^{2N} < p^{2(N+1)}$ for $p \geq 3$ in both two and three space dimensions ($N = 2, 3$), the flop count favors the implicit MTP scheme as p increases.

Further work is needed to provide rigorous proofs of the convergence rates for MTP schemes and to provide computational benchmarks for specific applications. Explicit MTP schemes are particularly interesting in the context of emerging many-core architectures, where data locality is important. More work is needed to realize these promises of improved performance of explicit MTP schemes, due to its better ratio of flops per memory (data locality) and matrix-free implementation techniques (such as sum factorization algorithms). Years of research on SDG schemes have resulted in advanced techniques like spacetime adaptive tent mesh refinement and element-wise conservation. Further studies are needed to bring such techniques to MTP schemes.

REFERENCES

[1] R. Abedi, B. Petracovici, and R. B. Haber, *A spacetime discontinuous Galerkin method for elastodynamics with element-wise momentum balance*, Computer Methods in Applied Mechanics and Engineering, 195 (2006), pp. 3247–3273.

[2] C. M. Dafermos, *Hyperbolic conservation laws in continuum physics*, vol. 325 of Grundlehren der Mathematischen Wissenschaften, Springer-Verlag, Berlin, third ed., 2010.

[3] J. Diaz and M. J. Grote, *Energy conserving explicit local time stepping for second order wave equations*, SIAM J. Sci. Comput., 31 (2009), pp. 1985–2014.

[4] J. Erickson, D. Guoy, J. M. Sullivan, and A. Üngör, *Building spacetime meshes over arbitrary spatial domains*, Engineering with Computers, 20 (2005), pp. 342–353.

[5] R. S. Falk and G. R. Richter, *Explicit finite element methods for symmetric hyperbolic equations*, SIAM J. Numer. Anal., 36 (1999), pp. 935–952.

[6] M. Gander and L. Halpern, *Techniques for locally adaptive time stepping developed over the last two decades*, in Domain Decomposition Methods in Science and Engineering XX, R. Bank, M. Holst, O. Widlund, and J. Xu, eds., vol. 91 of Lecture Notes in Computational Science and Engineering, Springer Berlin Heidelberg, 2013, pp. 377–385.

[7] J. Gopalakrishnan, P. Monk, and P. Sepúlveda, *A tent pitching scheme motivated by Friedrichs theory*, Computers and Mathematics with Applications, 70 (2015), pp. 1114–

1135.

[8] 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), pp. A747–A775.

[9] J.-L. Guermond, R. Pasquetti, and B. Popov, *Entropy viscosity method for nonlinear conservation laws*, J. Comput. Phys., 230 (2011), pp. 4248–4267.

[10] E. Hairer and G. Wanner, *Solving ordinary differential equations. II*, vol. 14 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1991. Stiff and differential-algebraic problems.

[11] G.-S. Jiang and E. Tadmor, *Nonoscillatory central schemes for multidimensional hyperbolic conservation laws*, SIAM J. Sci. Comput., 19 (1998), pp. 1892–1917 (electronic).

[12] R. B. Lowrie, P. L. Roe, and B. van Leer, *A space-time discontinuous Galerkin method for the time-accurate numerical solution of hyperbolic conservation laws*, in Proceedings of the 12th AIAA Computational Fluid Dynamics Conference, no. 95-1658, 1995.

[13] J. C. Mandal and S. M. Deshpande, *Kinetic flux vector splitting for Euler equations*, Comput. & Fluids, 23(2) (1994), pp. 447–478.

[14] S. T. Miller and R. B. Haber, *A spacetime discontinuous Galerkin method for hyperbolic heat conduction*, Computer Methods in Applied Mechanics and Engineering, 198 (2008), pp. 194–209.

[15] P. Monk and G. R. Richter, *A discontinuous Galerkin method for linear symmetric hyperbolic systems in inhomogeneous media*, J. Sci. Comput., 22/23 (2005), pp. 443–477.

[16] A. D. Mont, *Adaptive unstructured spacetime meshing for four-dimensional spacetime discontinuous Galerkin finite element methods*, Master's thesis, University of Illinois at Urbana-Champaign, 2011.

[17] M. Neumüller, *Space-Time Methods: Fast Solvers and Applications*, PhD thesis, Graz University of Technology, 2013.

[18] J. Palaniappan, R. B. Haber, and R. L. Jerrard, *A spacetime discontinuous Galerkin method for scalar conservation laws*, Computer Methods in Applied Mechanics and Engineering, 193 (2004), pp. 3607–3631.

[19] G. R. Richter, *An explicit finite element method for the wave equation*, Appl. Numer. Math., 16 (1994), pp. 65–80. A Festschrift to honor Professor Robert Vichnevetsky on his 65th birthday.

[20] J. Schöberl, *C++11 implementation of finite elements in NGSolve*, To appear, (2016).

[21] D. Serre, *Systems of conservation laws. 1*, Cambridge University Press, Cambridge, 1999. Hyperbolicity, entropies, shock waves, Translated from the 1996 French original by I. N. Sneddon.

[22] A. Üngör and A. Sheffer, *Pitching tents in space-time: mesh generation for discontinuous Galerkin method*, Internat. J. Found. Comput. Sci., 13 (2002), pp. 201–221. Volume and surface triangulations.

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

[24] L. Wang and P.-O. Persson, *A high-order discontinuous Galerkin method with unstructured space-time meshes for two-dimensional compressible flows on domains with large deformations*, Comput. & Fluids, 118 (2015), pp. 53–68.

[25] C. Wintersteiger, *Mapped tent pitching method for hyperbolic conservation laws*, Master's thesis, Technical University of Vienna, 2015.

[26] P. Woodward and P. Colella, *The numerical simulation of two-dimensional fluid flow with strong shocks*, J. Comput. Phys., 54 (1984), pp. 115–173.

[27] L. Yin, A. Acharia, N. Sobh, R. B. Haber, and D. A. Tortorelli, *A spacetime discontinuous Galerkin method for elastodynamics analysis*, in Discontinuous Galerkin Methods: Theory, Computation and Applications, B. Cockburn and G. Karniadakis and C. W. Shu(eds), Springer Verlag, 2000, pp. 459–464.