Construction of 4D Simplex Space-Time Meshes for Local Bisection Schemes

David Lenz

1 Introduction

Space-time finite element methods (FEMs) approximate the solution to a PDE 'all-at-once' in the sense that a solution is produced at all times of interest simultaneously. This is achieved by treating time as just another variable and discretizing the entire space-time domain with finite elements. Naturally, discretizing the entire space-time domain creates a linear system with many more degrees of freedom (DOFs) than discretizing just the spatial domain. Adaptive mesh refinement can produce space-time discretizations that yield accurate solutions with relatively few degrees of freedom. For example, Langer & Schafelner [4] compared space-time FEMs using uniformly and adaptively refined meshes; they found that obtaining the same approximation error for their tests required more degrees of freedom on the uniform meshes by one to two orders of magnitude.

A technical challenge in the implementation of adaptive mesh refinement is, of course, the mesh refinement scheme. In order to refine a geometric element while introducing as few new DOFs as possible, algorithms typically employ a "red-green" approach (uniform refinement with closure) or an element bisection approach. Here we focus on bisection schemes in four dimensions but note that work on arbitrary-dimensional red-green schemes was recently undertaken by Grande [3]. Stevenson [7] has studied a bisection algorithm for arbitrary-dimensional simplicial meshes, which is a good candidate for four-dimensional space-time meshes. However, the algorithm relies on a mesh precondition that is difficult to satisfy.

In this article, we weaken this strict precondition for certain space-time meshes. We prove that the precondition on four-dimensional simplex meshes can be reduced to a precondition on an underlying three-dimensional mesh. This means that the condition needs only be checked on a much smaller mesh. In addition, if one cannot

David Lenz
Argonne National Laboratory, Lemont, IL, USA, e-mail: `dlenz@anl.gov`

immediately verify the precondition, refinements to this smaller three-dimensional mesh can be made to automatically satisfy the precondition.

In section 2, we describe a particular method for creating four-dimensional space-time meshes. Meshes of this type have a structure which will be exploited in section 3, where we summarize key concepts from Stevenson's bisection algorithm and then prove our main result. Finally, we conclude with some remarks on how this relaxed precondition can be used in practice.

2 Four-Dimensional Space-Time Mesh Construction

When applying space-time FEMs to solve a PDE, it is generally necessary to create a space-time mesh that corresponds to a given spatial domain. A convenient method for doing so is to repeatedly extrude spatial elements (typically triangles or tetrahedra) into higher-dimensional space-time prisms and then subdivide these prisms into simplicial space-time elements (tetrahedra or pentatopes). We refer to mesh generation methods of this type as *extrusion-subdivision* schemes.

The method of extrusion-subdivision has appeared in several places in recent years. This idea was applied to moving meshes by Karabelas & Neumüller [6] and is discussed in a report by Voronin [9], where it is described in the context of an extension to the MFEM library [1]. For stationary (non-moving) domains, the extrusion step is straightforward, but subdividing the space-time prisms can be done in several ways. Behr [2] describes a method for subdividing space-time prisms using Delaunay triangulations, while subdivision based on vertex orderings is used in [6], [9].

In this paper, we consider space-time meshes produced by extrusion-subdivision where prism subdivision is defined in terms of vertex labels. In particular, we will assume that a k-coloring has been imposed on the mesh; that is, each vertex in the mesh has one of k labels (colors) attached to it and no two vertices connected by an edge share the same label.

Before we describe our particular prism subdivision method, we need to establish some notation. Let d be the spatial dimensionality of the problem and suppose $a = (a_0, a_1, \ldots, a_{d-1}) \in \mathbb{R}^d$. For any $r \in \mathbb{R}$, we define the map $\psi_r : \mathbb{R}^d \to \mathbb{R}^{d+1}$ by

$$\psi_r(a) = (a_0, a_1, \ldots, a_{d-1}, r). \tag{1}$$

For a simplex $T = \text{conv}(a, b, c, d)$ (here $a, b, c, d \in \mathbb{R}^d$), we define

$$\psi_r(T) = \text{conv}(\psi_r(a), \psi_r(b), \psi_r(c), \psi_r(d)); \tag{2}$$

that is, $\psi_r(T)$ is the embedding of T into $\mathbb{R}^{d+1}$ space-time within the plane $x_{d+1} = r$.

Next, the extrusion operator $\Phi_{r,s}$ produces the set of all points between $\psi_r(T)$ and $\psi_s(T)$. This is the convex hull of points in $\psi_s(T)$ and $\psi_r(T)$, which is a right tetrahedral prism:

$$\Phi_{r,s}(T) = \text{conv}(\psi_r(T), \psi_s(T)). \tag{3}$$

We make one final notational definition to declutter the following exposition. Given a series of real values $\mathcal{S} = \{s_0, \ldots, s_M\}$, we define

$$\Phi_i^{\mathcal{S}}(T) = \Phi_{s_i, s_{i+1}}(T) \quad \text{where } s_i \in \mathcal{S}, \text{ for } 0 \leq i \leq M-1. \tag{4}$$

We refer to $\mathcal{S}$ as a collection of "time-slices", which determine how spatial elements are extruded into space-time prisms. The value s_i is thus the "i^{th} time-slice". For most problems, the set of initial time-slices is fixed ahead of time, so it is often convenient to omit the superscript $\mathcal{S}$. We adopt this shorthand for the remainder of this paper. For an illustration of the operators ψ and Φ in two spatial dimensions, see figure 1.

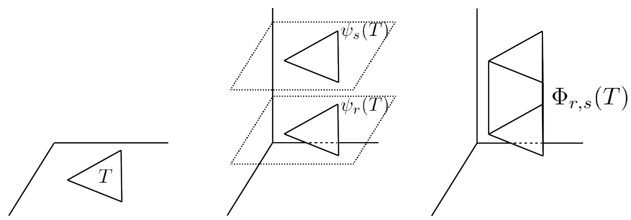

Fig. 1: Extrusion of a 2D simplex into a 3D simplex prism. Left: The spatial element. Center: Copies of the spatial element embedded in space-time. Right: The space-time prism element.

For the remainder of this article, we will focus on the case $d = 3$; that is, problems in four-dimensional space-time.[1] Let $\mathcal{T}$ be a conforming tetrahedral mesh with a 4-coloring,[2] and let the symbols A, B, C, D denote the four labels to be associated with each vertex of $\mathcal{T}$. Let $\mathcal{S}$ be a set of time-slices which define the extrusion of spatial elements into space-time prisms (cf. equation (4)). We will use the following rule to subdivide the tetrahedral prisms formed by extruding elements of $\mathcal{T}$.

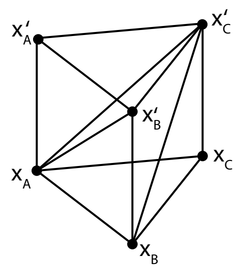

Fig. 2 Space-time triangular prism subdivision based on a coloring of the underlying spatial mesh.

[1] Due to the inherent difficulty in visualizing four-dimensional objects on a two-dimensional page, we will continue to illustrate figures in three-dimensional space-time. These figures are meant only as a guide for the reader to develop some geometric intuition.

[2] Not every tetrahedral mesh admits a 4-coloring, although all are 5-colorable. We discuss how to handle meshes which are not 4-colorable at the end of section 3.

Definition 1 (Subdivision Rule)

Let $T \in \mathcal{T}$ be a tetrahedron with vertices v_A, v_B, v_C, v_D, where the subscript of each vertex denotes its color label. For a given time slice $s_i \in \mathcal{S}$, let $x_A = \psi_{s_i}(v_A)$ and $x'_A = \psi_{s_{i+1}}(v_A)$ (and likewise for B, C, D). The rule for subdividing $\Phi_i(T)$ is to create the pentatopes:

$$\begin{aligned}\tau_1 &= \mathrm{conv}(x_A, x_B, x_C, x_D, x'_D)\\ \tau_2 &= \mathrm{conv}(x_A, x_B, x_C, x'_C, x'_D)\\ \tau_3 &= \mathrm{conv}(x_A, x_B, x'_B, x'_C, x'_D)\\ \tau_4 &= \mathrm{conv}(x_A, x'_A, x'_B, x'_C, x'_D)\end{aligned} \tag{5}$$

An illustration of this subdivision rule is given in figure 2. Because the labeling of each vertex is shared by all elements, this subdivision scheme always produces a conforming mesh of pentatopes. However, we omit the detailed proof because of space constraints.

3 Conforming Bisection of Space-Time Simplicial Elements

The aim of this section is to outline Stevenson's bisection algorithm [7][3], with particular attention on the mesh precondition for its validity. Then, we show that space-time meshes produced by extrusion-subdivision (using definition 1) will always meet this precondition if the underlying spatial mesh is 4-colorable. The upshot of this is that all of the work to make an admissible mesh can be done in three dimensions instead of four. This is especially useful in light of the fact that there are many more meshing utilities for three-dimensional domains than four-dimensional domains.

The following definitions are due to Stevenson [7].

Definition 2 A *tagged pentatope* t is an ordering of the vertices of some pentatope $\tau = \mathrm{conv}(x_0, x_1, x_2, x_3, x_4)$, together with an integer $0 \le \gamma \le 3$ called the *type*. We write

$$t = (x_0, x_1, x_2, x_3, x_4)_\gamma \tag{6}$$

to denote this ordering-type pair.

Definition 3 The *reflection* of a tagged pentatope t is another tagged pentatope t_R such that the bisection rule produces the same child pentatopes for t and t_R. The unique reflection of $t = (x_0, x_1, x_2, x_3, x_4)_\gamma$ is

$$t_R = \begin{cases} (x_4, x_3, x_2, x_1, x_0)_\gamma & \text{if } \gamma = 0 \\ (x_4, x_1, x_3, x_2, x_0)_\gamma & \text{if } \gamma = 1 \\ (x_4, x_1, x_2, x_3, x_0)_\gamma & \text{if } \gamma = 2, 3 \end{cases} . \tag{7}$$

[3] The bisection rule studied by Stevenson has also been studied by Maubach [5] and Traxler [8].

Definition 4 Two tagged pentatopes t and t' are *reflected neighbors* if they share a common hyperface, have the same integer type, and the vertex order of t' matches the vertex order of t or t_R in all but one position.

The notion of reflected neighbors is critical to our proof of the main result, so it is worth illustrating the concept with some examples.

Example 1 Let $t = (x_0, x_1, x_2, x_3, y)_1$ and $t' = (z, x_0, x_1, x_2, x_3)_1$. Then t and t' are NOT reflected neighbors. Although the relative ordering of their shared vertices is consistent, the taggings differ in every position.

Example 2 Let $t = (x_0, z, x_1, x_2, x_3)_1$ and $t' = (x_3, y, x_2, x_1, x_0)_1$. Then t and t' are reflected neighbors. To see this, we note that $t_R = (x_3, z, x_2, x_1, x_0)_1$; thus t_R and t' differ only in the second position. Likewise, we could have shown that t'_R and t differ on at most one position.

Definition 5 The tagging of two pentatopes $t = (x_0, x_1, x_2, x_3, x_4)_\gamma$ and $t' = (x'_0, x'_1, x'_2, x'_3, x'_4)_\gamma$ which share a hyperface is said to be *consistent* if the following condition is met:

1. If $\overline{x_0, x_4}$ or $\overline{x'_0, x'_4}$ is contained in the shared hyperface, then t and t' are reflected neighbors (N.B. these are the bisection edge for each element; see definition 6).
2. Otherwise, the two children of t and t' which share the common hyperface are reflected neighbors.

A *consistent tagging of a mesh* is a tag for each element in the mesh such that any two neighboring elements are consistently tagged.

In essence, definition 5 states that in a consistently tagged mesh, any pair of neighboring elements are either reflected neighbors or they do not share a common refinement edge. Furthermore, when two elements do not share a common refinement edge, their adjacent children will be reflected neighbors after one round of bisection.

Definition 6 (Bisection Rule)

Given a tagged pentatope $t = (x_0, x_1, x_2, x_3, x_4)_\gamma$, applying the bisection rule produces the children:

$$t_1 = (x_0, x', x_1, x_2, x_3)_{\gamma'} \qquad t_2 = \begin{cases} (x_4, x', x_3, x_2, x_1)_{\gamma'} & \text{if } \gamma = 0 \\ (x_4, x', x_1, x_3, x_2)_{\gamma'} & \text{if } \gamma = 1 \\ (x_4, x', x_1, x_2, x_3)_{\gamma'} & \text{if } \gamma = 2, 3 \end{cases}, \tag{8}$$

where $x' = (x_0 + x_4)/2$ and $\gamma' = (\gamma + 1) \mod 4$. We say that edge $\overline{x_0 x_4}$ is the *refinement edge*.

We can now state the main result of this section.

Proposition 1 *Let $\mathcal{T} \subset \mathbb{R}^3$ be a 4-colorable tetrahedral mesh and $\mathcal{T}' \subset \mathbb{R}^4$ be the pentatope mesh produced by extrusion-subdivision according to definition 1. Then $\mathcal{T}'$ admits a consistent tagging.*

Proof We will prove the result by constructing a consistent tagging of $\mathcal{T}'$ directly from a 4-coloring of $\mathcal{T}$. Every simplex is tagged according to its position within its extruded space-time prism from definition 1. For the above four pentatopes in definition 1, we make the following tagging (in each case t_i is a tagging of τ_i):

$$\begin{aligned} t_1 &= (x_D, x_C, x_B, x_A, x'_D)_0 \qquad & t_2 &= (x_C, x_B, x_A, x'_D, x'_C)_0 \\ t_3 &= (x_B, x_A, x'_D, x'_C, x'_B)_0 \qquad & t_4 &= (x_A, x'_D, x'_C, x'_B, x'_A)_0 \end{aligned} \tag{9}$$

To show that a tagging of $\mathcal{T}'$ is consistent, it suffices to consider an arbitrary element of $\mathcal{T}'$ and show that each of its neighbors satisfy the conditions in definition 5. Let $\tau \in \mathcal{T}'$ be an abitrary element. Since $\mathcal{T}'$ is created by extrusion-subdivision, there is some time-slice s_i and $T \in \mathcal{T}$ such that $\tau \subset \Phi_i(T)$.

We will show that each neighbor τ' satisfies the consistency condition in definition 5. There are three cases, illustrated in figure 3:

1. τ and τ' are both pentatopes within the same space-time prism.
2. τ and τ' belong to different space-time prisms extruded from the same spatial element; for instance, $\tau \subset \Phi_i(T)$ and $\tau' \subset \Phi_{i+1}(T)$.
3. τ and τ' belong to different space-time prisms within the same space-time slab; for instance, $\tau \subset \Phi_i(T)$ and $\tau' \subset \Phi_i(T')$.

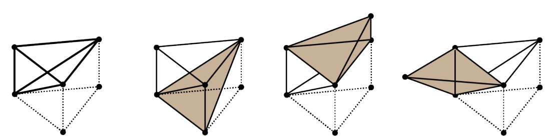

Fig. 3 Different types of neighboring elements. The three shaded elements are the three types of neighbors for the bold simplex given at left.

We verify consistency of the tagging scheme with repeated applications of definition 1 and few geometric arguments, which ensures that all cases are covered.

Consider Case (1). Since τ and τ' are neighbors, they must be a pair τ_i, τ_{i+1} ($i = 1, 2, 3$) from definition 1, since only consecutive pairs in our list are neighbors. In this case, the adjacent children of each pentatope are reflected neighbors. To make this point explicit, consider the neighboring tagged elements t_1, t_2. The children formed by bisecting these pentatopes are:

$$t_1 \rightarrow \begin{cases} (x_D, z_1, x_C, x_B, x_A)_1 \\ (x'_D, z_1, x_A, x_B, x_C)_1 \end{cases} \qquad t_2 \rightarrow \begin{cases} (x_C, z_2, x_B, x_A, x'_D)_1 \\ (x'_C, z_2, x'_D, x_A, x_B)_1 \end{cases} \tag{10}$$

where z_i are the new midpoints of the bisected edges. From here we note that the second child of t_1 is the reflected neighbor of t_2. The same exercise shows that the pairs t_2, t_3 and t_3, t_4 also share this property, and thus all are consistently tagged.

In Case (2), τ and τ' are neighbors in different space-time slabs; without loss of generality, $\tau \subset \Phi_i(T)$ and $\tau' \subset \Phi_{i+1}(T)$ for some $T \in \mathcal{T}$. With two space-time prisms, we have three sets of vertices at different time-slices. Denote vertices in the highest (latest) time hyperplane with double primes (”), the middle hyperplane with single primes (’), and the lowest (earliest) hyperplane with no primes; see figure 4.

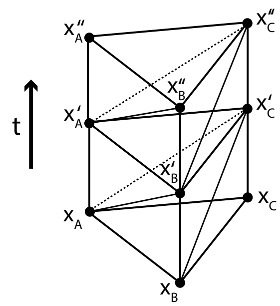

Fig. 4 Illustration of vertex labeling when considering consecutive space-time time prisms in two spatial dimensions.

Since τ and τ' belong to consecutive timeslices, the shared hyperface between the two must be $\mathrm{conv}(x'_A, x'_B, x'_C, x'_D)$. Thus $\tau = \mathrm{conv}(x_A, x'_A, x'_B, x'_C, x'_D)$ and $\tau' = \mathrm{conv}(x'_A, x'_B, x'_C, x'_D, x''_D)$. According to the tagging scheme described above, the tags on these two pentatopes are

$$t = (x_A, x'_D, x'_C, x'_B, x'_A)_0 \quad t' = (x'_D, x'_C, x'_B, x'_A, x''_D)_0, \tag{11}$$

and thus their child elements are

$$t \to \begin{cases} (x_A, z, x'_D, x'_C, x'_B)_1 \\ (x'_A, z, x'_B, x'_C, x'_D)_1 \end{cases} \qquad t' \to \begin{cases} (x'_D, z', x'_C, x'_B, x'_A)_1 \\ (x''_D, z', x'_A, x'_B, x'_C)_1 \end{cases}, \tag{12}$$

where z, z' are new vertices created by bisecting τ and τ'. This is indeed a consistent tagging, as the second child of t and the first child of t' are reflected neighbors.

Finally, consider Case (3). Since $\mathcal{T}'$ is conforming, when $\tau \subset \Phi_i(T)$ and $\tau' \subset \Phi_i(T')$ they must share a vertical edge like $\overline{x_A x'_A}$, which is always a bisection edge. Since vertex labels are "global" labels, both τ and τ' agree on the order in which the labeled vertices appear. Furthermore, τ and τ' share all but one vertex in common. Since all but one vertex is shared, both pentatopes have the same labels for the same vertices, and vertex order is uniquely determined by vertex label, the vertex orders agree on all but one position. Hence the tagged pentatopes are reflected neighbors.

Thus all three cases result in consistent taggings. Therefore, the tagging defined by equation (9) is consistent. □

The critical piece of this proof is the tagging scheme defined in equation (9). Furthermore, since the vertex orders are determined by the 4-coloring on $\mathcal{T}$, a consistent tagging of the space-time mesh can be constructed in linear time.

Corollary 1 *Let $\mathcal{T}$ and $\mathcal{T}'$ be as in proposition 1. Given a 4-coloring on $\mathcal{T}$, the space-time mesh $\mathcal{T}'$ can be consistently tagged in $O(N)$ time, where N is the number of vertices in $\mathcal{T}'$.*

Not every tetrahedral mesh is 4-colorable, but this can be worked around. First, we note that regular tetrahedral meshes are 4-colorable, so for rectilinear domains one can start with a coarse regular mesh and bisect until a desired resolution is met.

In addition, Traxler [8] has shown that tetrahedral meshes over simply connected domains are 4-colorable iff every edge is incident to an even number of tetrahedra.

Finally, any tetrahedral mesh can be made 4-colorable by dividing each element via barycentric subdivision and then choosing the following colors: each vertex of the original mesh is colored A, the new center of each edge is colored B, the new center of each face is colored C, and the new center of each tetrahedron is colored D. Barycentric subdivision creates new elements with one of each kind of point, which means that this is indeed a 4-coloring of the subdivided mesh.

4 Conclusions

We described a method for creating four-dimensional simplex space-time meshes from a given spatial mesh which has a 4-coloring. This procedure was based on the general extrusion-subdivision framework, with a new subdivision rule which is defined in terms of vertex labels (colors). We then proved that meshes of this form always satisfy the strict precondition of Stevenson's bisection algorithm, which can be used to adaptively refine space-time meshes. Finally, we showed that even when a tetrahedral mesh is not 4-colorable, the barycentric subdivision of the mesh will be.

Acknowledgements This work is supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research under Contract DE-AC02-06CH11357, and the Exascale Computing Project (Contract No. 17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

References

1. MFEM: Modular finite element methods library. `mfem.org`. DOI 10.11578/dc.20171025.1248
2. Behr, M.: Simplex space-time meshes in finite element simulations. Internat. J. Numer. Methods Fluids **57**(9), 1421–1434 (2008). DOI 10.1002/fld.1796
3. Grande, J.: Red-green refinement of simplicial meshes in d dimensions. Math. Comp. **88**(316), 751–782 (2019). DOI 10.1090/mcom/3383
4. Langer, U., Schafelner, A.: Adaptive space-time finite element methods for non-autonomous parabolic problems with distributional sources. Comput. Methods Appl. Math. **20**(4), 677–693 (2020). DOI 10.1515/cmam-2020-0042
5. Maubach, J.M.: Local bisection refinement for n-simplicial grids generated by reflection. SIAM J. Sci. Comput. **16**(1), 210–227 (1995). DOI 10.1137/0916014
6. Neumüller, M., Karabelas, E.: Generating admissible space-time meshes for moving domains in $(d+1)$ dimensions. In: U. Langer, O. Steinbach (eds.) Space-Time Methods, pp. 185–206. De Gruyter, Berlin, Boston (2019). DOI 10.1515/9783110548488-006
7. Stevenson, R.: The completion of locally refined simplicial partitions created by bisection. Math. Comp. **77**(261), 227–241 (2008). DOI 10.1090/S0025-5718-07-01959-X
8. Traxler, C.T.: An algorithm for adaptive mesh refinement in n dimensions. Computing **59**(2), 115–137 (1997). DOI 10.1007/BF02684475
9. Voronin, K.: A parallel mesh generator in 3d/4d. Tech. Rep. 11, Portland Institute for Computational Science Publications (2018)