Constrained 2D Space-Time Meshing With All Tetrahedra

Alper Üngör * Cinda Heeren * Xiang-Yang Li * Alla Sheffer † Robert Haber ‡
Shang-Hua Teng *

Abstract

Space-time discontinuous Galerkin (DG) methods provide a solution for elastodynamic analysis, a problem that serves as a model for DG approximations of second-order hyperbolic problems. A direct element-by-element solution motivates the generation of a two dimensional (2D) space-time mesh with a cone constraint, the faces connecting two time slabs can not be steeper than a specified angle α with respect to the 2D space. This paper presents a solution to this constrained 2D space-time meshing problem using only simplicial elements.

Key words: Mesh generation, space-time meshes, computational geometry, tetrahedral meshes, algorithms.

AMS subject classifications: 65D18, 65M50, 68U05, 68W01.

1 Introduction

Space-time meshing The space-time methods formulate the numerical simulation problem using finite element discretization of the time domain as well as the usual discretization of the spatial domain. They are often coupled with discontinuous Galerkin methods [4]. Usually the discretization of the space-time domain utilizes time slabs, as in [9]. Space-time methods prefer to employ unstructured meshes allowing for varying time resolution in different parts of the domain. In the last two decades, there has been wide research on space-time DG methods [4, 7, 9] However, none of the existing space-time meshing techniques is designed to handle the cone constraint.

Motivation for cone constraint Our motivation for studying space-time meshes with angle constraint is due to a new space-time DG method for elastodynamic analysis, introduced by Yin *et al.* [8]. The importance of the elastodynamic analysis problem is due to its being a model for DG approximations of second-order hyperbolic problems in general. The method of Yin *et al.* supports unstructured, fully-discontinuous space-time discretizations, enforces element-wise conservation and is compatible with direct element-by-element solution procedures. Hence there is no need to solve a large global system of equations. The key to this formulation is the identification of the proper conditions across an arbitrary surface of discontinuity embedded in a space-time domain [8]. The method depends on the notion of domains of influence for dynamic data [3]. Let $\mathcal{E}_{\mathbf{z}}$ be some elastodynamic event that occurs at $\mathbf{z} \in \mathcal{D}$. The domain of influence of $\mathcal{E}_{\mathbf{z}}$, denoted $\mathcal{I}_{\mathbf{z}}$, is the interior of a cone with vertex $\mathbf{z}$ and whose boundaries are determined by the fastest wave speed in the material. To satisfy this cone condition, the construction of a special 2D space-time mesh, which will be formally defined in the next section, is essential.

The DG analysis imposes no restricition on the mesh element shape, i.e. tetrahedral, hexahedral and mixed meshes possibly with other element shapes are acceptable. In a companion paper we present a hexahedral mesh generation scheme preserving the cone constraint [6]. In this paper, we have choosen to use simplicial elements (triangles and tetrahedra) as they are known to be more suitable than others discretizing complex geometries.

Outline Next section gives a formal definition of the problem. Section 3 gives a three step algorithm to solve the problem. The main theorem of this paper which states that the given algorithm generates a space-time mesh that achieves the angle constraint is proven in section 4. We conclude with some remarks on mixed element meshes, boundary elements and ideas on how to get a better angled mesh.

*Department of Computer Science, University of Illinois at Urbana-Champaign, corresponding author's e-mail: ungor@cs.uiuc.edu
†Computational Science and Engineering, University of Illinois at Urbana-Champaign
‡Department of Theoretical and Applied Mechanics, University of Illinois at Urbana-Champaign

2 Problem Definition

To construct a space-time mesh that achieves the cone condition we can consider the following single layer meshing problem with the angle constraint.

Definition 2.1 [Single time slab problem] Given a 2D unstructured mesh, construct a single layer of space-time mesh such that for each face f of the mesh, the dihedral angle between f and the 2D space is less than a specified angle α (usually 45°). ■

In this definition our major concern is the faces between time slabs, as the other faces give a zero (or small if we don't have the time slabs parallel) angle. Once we have a solution for the single time slab problem, we can use it to solve the general space-time mesh problem. The solution for each consecutive time slab can be computed by flipping over the mesh from the previous time slab and modifying it to conform to the new boundary. Specifically, let m_i be the 2D space mesh used at time t_i, and s_i and m_{i+1} be the solution of the single time slab problem, where m_{i+1} is the space mesh constructed for the time t_{i+1} and s_i defines the connections between m_i and m_{i+1}. By flipping this solution over and adjusting it near the boundary, we can provide a solution for the next slab. In other words this solution suggests using m_i for m_{i+2} and mirror image of s_i for s_{i+1} .

Achieving the cone constraint on the boundary and preserving the geometry of the domain boundary are two conflicting objectives. The angles of the faces that are incident to the boundary are imposed by the geometry of the boundary. Hence, faces that are incident to the boundary are not required to satisfy the cone constraint.

Notation For a triangle element, $\triangle abc$, at layer i, let $\bigcirc abc$ denote the largest circle (insphere) inside the triangle. Let p_{abc} be the center of $\bigcirc abc$. The projection of this point to the next layer $(i+1)$, which will be computed by the step 1 of the algorithm `ATSTM` is denoted by p'_{abc} (see Figure 2). Let p_{bc} be the point where line bc, intersects the line from p_{abc} to p_{bcd} (see Figure 3). Also $p_{a\overline{bc}}$ is used to denote the point where $\bigcirc abc$ touches the edge bc.

3 Algorithm

An all tetrahedra solution requires an initial (time t_0) triangular mesh for the space domain. Specifically we use a non-obtuse triangular mesh which can be generated in polynomial time by one of the algorithms given in [1, 2].

Algorithm ATSTM is designed to generate a space time mesh with angle constraint in the interior of the domain. In the design of the algorithm our major concern is achieving the angle constraint for the faces between two spatial mesh layers. Each of those faces has one vertex on one mesh (say on m_i) and two on the other (m_{i+1}) or vice versa. To minimize the face angle, one should maximize the distance between the projection of the single vertex to the other mesh layer and the line passing through the other two vertices that are already on the same mesh layer. Using non-obtuse triangulation supports this objective.

Generation of the elements incident to the boundary which do not need to conform the angle constraint is discussed in Section 5.

Algorithm `All Tetrahedra Space-Time Meshing (ATSTM)`

1. [**Face based tets**]: For every triangle $\triangle abc$ in m_i (two-dimensional space mesh at time t_i), compute the center of the inscribed sphere, p_{abc}. Projecting this point to the next layer, compute p'_{abc} as a mesh vertex of m_{i+1}. Using p'_{abc} as the apex, construct the tetrahedra $abcp'_{abc}$. See Figure 1 (b).

2. [**Edge based tets**]: In between every two neighbor tetrahedra formed in step 1 that share an edge, introduce a new tetrahedron by putting an edge between their apexes. Each one of these new tetrahedra introduces two new faces. This step results in polygon based pyramid shaped cavities. See Figure 1 (c).

3. [**Vertex based tets**]: Let ap be a cavity which resulted from the previous step, where a is the apex of the cavity that is on m_i and p is the polygon on m_{i+1}. Let a' be the projection of a on p. Find the triangle in p that covers the point a' best, meaning maximizes the minimum distance of a' to an edge of the triangle. Triangulate p including this specific best covering triangle. Fill the cavity by introducing the tetrahedra using the simplices in the triangulation of p as the bases and the point a as the apex. See Figure 1 (d).

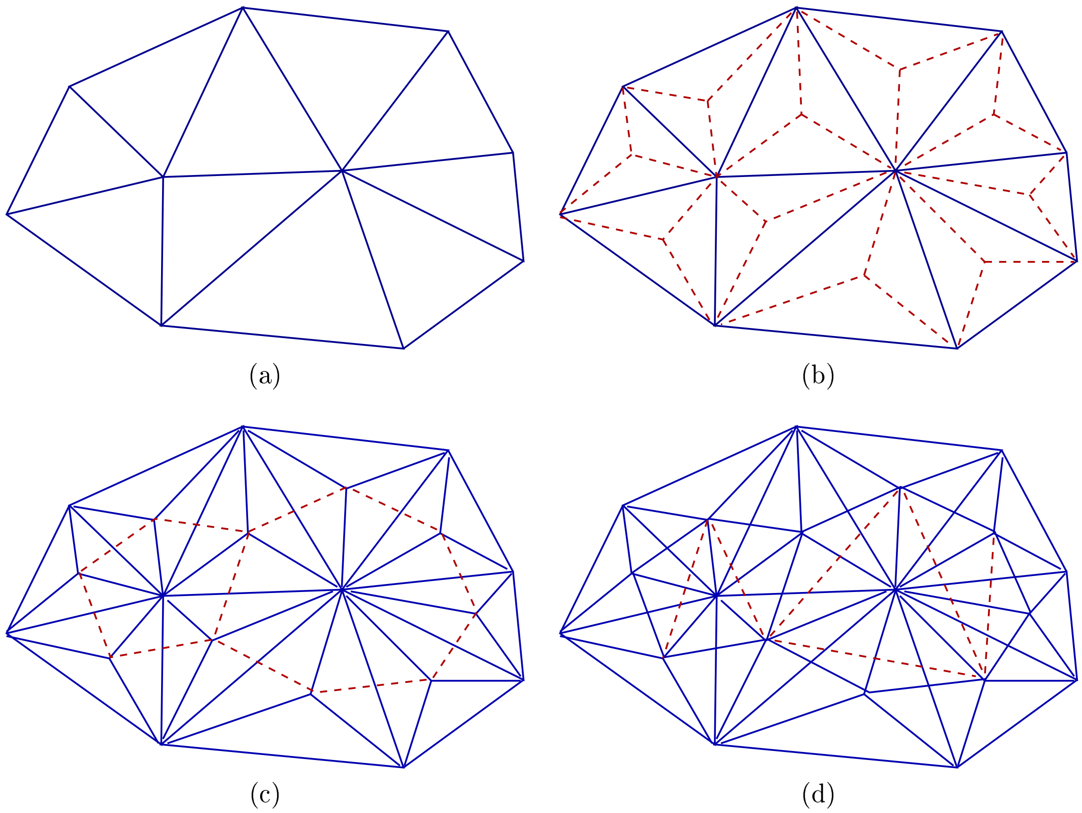

Figure 1: Sketch of the algorithm. (a) 2D space mesh m_i (b) step 1; face based tets (c) step 2; edge based tets (d) step 3; vertex based tets

4 Achieving the Angle Constraint

Lemma 4.1 *Face angles introduced in the first step of Algorithm* ATSTM *are optimal, i.e. no other face based tetrahedra with nonobtuse-dihedral angle can acheive a larger height for angle constraint α.*

Proof Note that the nonobtuse dihedral angle constraint of the face based tetrahedra essentially allows the projection of the apexes only inside the base triangles. Relaxing this constraint deteriorates the mesh quality dramatically. Let p_{abc} be the projection of the apex, p'_{abc}, of a face base tetrahedra whose face triangle is $\triangle abc$. See Figure 2. The face angles, dihedral angles, introduced by the three new faces, $p'_{abc}ab$, $p'_{abc}bc$, and $p'_{abc}ac$ are directly related to how far the point p_{abc} is from the edges ab, bc, and ac, respectively. The minimum of these distances is maximized simply by choosing the point p_{abc} as the center of the insphere $\bigcirc abc$. ■

Lemma 4.2 *Face angles introduced in the second step of Algorithm* ATSTM *are no worse than the face angles introduced in the first step.*

Proof Each tetrahedra generated in step 2, has two old faces constructed in step 1 and two new faces. Here we show that a new face has dihedral angle not larger than the steepest angle incident old faces of the two neighbor tetrahedra. In Figure 3, we claim that the angle introduced by the face $\triangle cp'_{abc}p'_{bcd}$ is no worse than the worst of the two faces $\triangle acp'_{abc}$ and $\triangle cdp'_{bcd}$. Let x be the closest point on the line defined by the points p_{abc} and p_{bcd} to c. To prove the claim, it is sufficient to show that

$$|cx| \geq min(|p_{abc}p_{\overline{bac}}|, |p_{bcd}p_{\overline{bcd}}|)$$

Since the angle $\widehat{acd}$ is less than 180°, the line defined by the points p_{abc} and p_{bcd} intersects either the line defined by the points a and c or line defined by c and d or both.

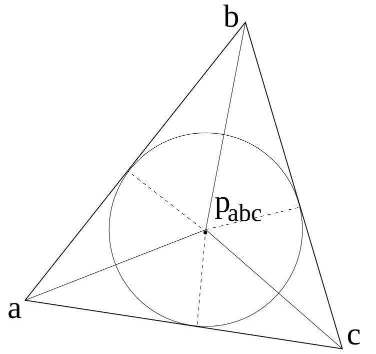

Figure 2: Step 1 of the Algorithm ATSTM optimizes the angle by using the center of the insphere.

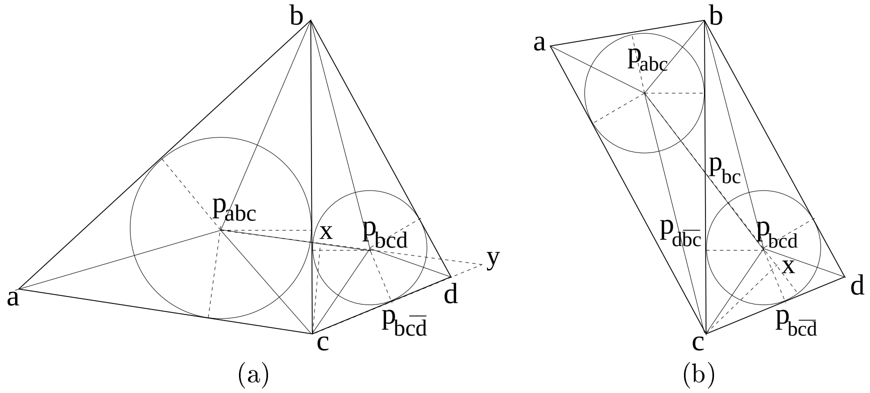

Figure 3: Step 2 of the Algorithm ATSTM introduces angles no worse than the step 1.

Notice that point p_{bc} divides the edge bc into two pieces each of which includes one of the tangent points $p_{d\overline{bc}}$ and $p_{a\overline{bc}}$. In the degenarate case, where bc is perpendicular to $p_{abc}p_{bcd}$, these two points, $p_{d\overline{bc}}$ and $p_{a\overline{bc}}$, are same. In this special case proof is easy. Consider the triangle $\triangle xcp_{bcd}$. The angle $\widehat{bcd} \leq 90°$ because of non-obtuse triangulation. This implies $\widehat{xcp_{bcd}} \leq 45°$ because cp_{bcd} is the angle bisector. This in turn implies that

$$|cx| \geq |p_{bc}p_{bcd}| = |p_{bcd}p_{b\overline{cd}}|$$

With no loss of generality, suppose $p_{d\overline{bc}}$ is closer to c, that is $|cp_{bc}| \geq |cp_{d\overline{bc}}|$. This implies $\widehat{cp_{bc}p_{bcd}} \leq 90°$. Since $\widehat{bcd}$ is also less than 90°, the two rays $\overrightarrow{p_{bc}p_{bcd}}$ and $\overrightarrow{cd}$ intersect. Let this intersection point be y.

There are two cases for the location of point x. It can be either on the line segment $p_{bc}p_{bcd}$ (see Figure 3 (a)) or on the other part of the ray $\overrightarrow{p_{bc}p_{bcd}}$ (see Figure 3 (b)).

In the first case, Figure 3 (a), consider the two triangles $\triangle xyc$ and $\triangle p_{b\overline{cd}}yp_{bcd}$. They are similar, because two of their angles (the right angle and the common angle $\widehat{xyc}$) are same. Since the triangle $\triangle p_{b\overline{cd}}yp_{bcd}$ is contained inside the triangle $\triangle xyc$, following relation between the two corresponding sides can be established:

$$|cx| \geq |p_{bcd}p_{b\overline{cd}}|$$

In the second case, Figure 3 (b), consider the two triangles,$\triangle xp_{bc}c$ and $\triangle p_{d\overline{bc}}p_{bc}p_{bcd}$. They are similar, because two of their angles (the right angle and the common angle $\widehat{xp_{bc}c}$) are same. Since the triangle $\triangle p_{d\overline{bc}}p_{bc}p_{bcd}$ is contained inside the triangle $\triangle xp_{bc}c$ following relation between the two corresponding sides can be established:

$$|cx| \geq |p_{d\overline{bc}}p_{bcd}|$$

Also $|p_{d\overline{bc}}p_{bcd}| = |p_{bcd}p_{b\overline{cd}}|$ both being the radius of the insphere $\bigcirc bcd$. Hence,

$$|cx| \geq |p_{bcd}p_{b\overline{cd}}|$$

■

Lemma 4.3 *Faces introduced in the third step of the Algorithm* `ATSTM`*, have a constant angle bound.*

Proof The most critical tetrahedra is the one formed by the best covering triangle, as all the others introduce faces with smaller angles. For polygons that has more than four points we can prove that a best covering triangle that includes the point a' in the interior exists. The proof goes through a case anylsis by considering the location of point a' with respect to the diagonals of all possible triangulations of this polygon. If point a' is in not on any of the diagonals the existence of a covering triangle is immediate. Otherwise if a' is on a diagonal or on the intersection of two or more diagonals one can always find a covering triangle by checking using the points that are on the regions divided by the diagonals. For 4 points a best covering triangle does not exist only if the point a' is at the intersection of the two diagonals. In that case we can perturb a' as we have enough room for it. ∎

Theorem 4.1 *Algorithm* `ATSTM` *generates a layer of mesh that conforms the specified α angle constraint everywhere except the boundary faces.*

Proof An upper bound of angle strictly less than 90° is sufficient to prove the theorem, as we can decrease that bound by decreasing the height of the time slab. To avoid vertical faces, the algorithm basically avoids introducing a triangular face based on an edge e and a vertex v, where v's projection to the opposite layer is on e. All tetrahedra introduced in the first step has their apex inside the base triangle and away from the edges hence upper bound of the angle is strictly less than 90°. Every edge introduced between a pair of neighbor apexes in the second step intersect its corresponding shared edge (between the base faces of the apexes) at an interior point, because we have non-obtuse triangulation assumption for mesh m_i. So the new faces also satisfy the angle constraint. Indeed Lemma 2 claimed a stronger result, that is the angles introduced in the second step is no worse than the ones in the first step. Finally, the third step is also safe due to Lemma 3. ∎

5 Boundary Elements

To complete the description of our angle constrained time-space meshing method, this section discusses the generation of elements incident to the boundary. Due to our assumption, angle constraint is not required for the boundary faces. After the execution of Algorithm `ATSTM`, we have a layer of cavities along the boundary.

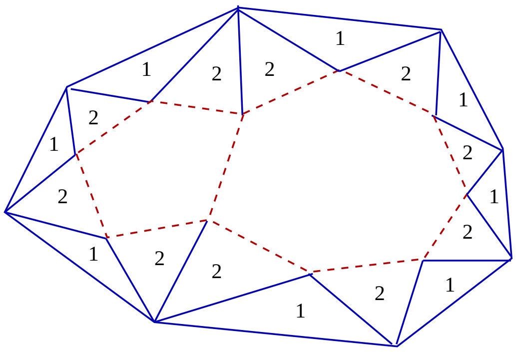

Figure 4: Algorithm `ATSTM` causes cavities along the boundary. Interior points are located one layer higher than the boundary points. Seperating these cavities by vertical faces gives us two types of geometric shapes within the time slab: pyramids labelled with 1 and tetrahedra labelled with 2.

These layers can be discretized into cells using vertical faces along the edges that are incident to the boundary. These cells can be classified into two: (i) the ones that are based on a triangle based tetrahedra face (ii) the ones that based on an edge based tetrahedra face. The second type of cells are tetrahedra already. The first types are pyramids whose quadrilateral faces are completely on the boundary. One can divide this cell into two tetrahedra by cutting the quadrilateral face.

6 Final Remarks

Height of the time slab We should note that the height of the time slab is closely related with the smallest feature size of the mesh.

Smoothing Also relaxing the criteria used to choose the mesh vertices of m_{i+1} in step 1, one can achieve better height with the same angle bound. In other words perturbing the mesh vertices and hence deteroiating the angle bound achieved in step 1, we can improve the angle bound in other steps and overall.

Polygonal based pyramid elements One can omit the third step of Algorithm ATSTM and generate a mixed element mesh. As discussed in Section 4 step 3 of the algorithm achieves the weakest angle bound. So having two step algorithm can generate a mesh that achieves larger time steps for a specified angle α. This can be coupled with a solver that handles polygonal based pyramid elements.

7 Acknowledgments

The work reported here was supported, in part, by the Center for Process Simulation and Design (NSF DMS 98-73945) and the Center for Simulation of Advanced Rockets (DOE LLNL B341494). The authors would like to thank Herbert Edelsbrunner, Jeff Erickson and John Sullivan for fruitful discussions during the course of this paper.

References

[1] M. Bern and D. Eppstein *Polynomial-size nonobtuse triangulation of polygons,* Int. J. Comp. Geometry and Applications,(2), 241-255, (1992).

[2] M. Bern, S. Mitchell and J. Ruppert *Linear-Size Nonobtuse Triangulation of Polygons,* Proc. of 10^{th} Symp. on Computational Geometry, New York, 221-230, (1994).

[3] M. E. Gurtin *The Linear Theory of Elasticity*, Handbuch der Physik 6a$_2$, Springer-Verlag, Berlin (1972).

[4] T. J. R. Hughes and G. M. Hulbert *Space-Time Finite Element Methods for Elastodynamics: Formulations and Error Estimates,* Computer Methods in Applied Mechanics and Engineering, 66:339-363, (1988).

[5] R. B. Lowrie, P. L. Roe and B. van Leer, *Space-Time Methods for Hyperbolic Conservation Laws*, Barriers and Challenges in Computational Fluid Dynamics, Kluwer, (6), 79-98, (1998),

[6] A. Sheffer, A.Üngör, S.-H. Teng and R. B. Haber *Generation of 2D Space-Time Meshes Obeying the Cone Constraint* To appear in Proc. of International Conference on Computational Engineering Science, Los Angeles, CA, (2000).

[7] L. L. Thompson *Design and Analysis of Space-Time and Galerkin Least-Squares Finite Element Methods for Fluid-Structure Interaction in Exterior Domains,* Ph.D. thesis, Stanford University, (1994).

[8] L. Yin, A. Acharya, N. Sobh, R. B. Haber and D. A. Tortorelli *A Space-Time Discontinuous Galerkin Method for Elastodynamic Analysis,* Proc. of International Symp. on Discontinuous Galerkin Methods, Salve Regina University, Newport, RI, May 24-26,(1999).

[9] P.J. Zwart and G.D. Raithby *Space-Time Meshing for Two-Dimensional Moving Boundary Problems,* Proc. of 7^{th} International Meshing Roundtable, Dearborn, MI, 187-200, (1998).