Generation of 2D Space-Time Meshes Obeying the Cone Constraint

Alla Sheffer, Alper Üngör, Shang-Hua Teng
Department of Computer Science,
University of Illinois at Urbana-Champaign

Robert Haber
Department of Theoretical and Applied Mechanics,
University of Illinois at Urbana-Champaign

Summary

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 three dimensional (2D×time) space-time meshes that satisfy a special cone constraint, which requires that the angle between each interior mesh face and the 2D space does not exceed a specified angle α.

The cone constraint precludes the use of existing meshing techniques for generating the 3D space-time mesh, so new approaches are needed. This paper provides a new algorithm for generating hexahedral meshes that satisfy the cone constraint.

Introduction

The analysis domain changes over time in many numerical simulation problems. Thus the mesh of the domain must be adaptively and dynamically updated to track the evolving geometry. Generation of space-time meshes is an alternative to mesh adaptation, where time is introduced as an additional dimension. This converts the problem of adaptive mesh modification into a mesh generation problem.

A new space-time discontinuous Galerkin (DG) method for elastodynamic analysis is introduced by Yin *et al.* [1]. The elastodynamic analysis problem is important since it serves as 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 it eliminates the need to solve a large global system of equations and provides an $O(n)$ algorithm. A key to the formulation is the identification of the proper conditions across an arbitrary surface of discontinuity embedded in a space-time domain [1]. The method depends on the notion of domains of influence for dynamic data [2]. 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 $\mathbf{z}$ whose boundaries are determined by the fastest wave speed in the material.

Direct element-by-element solutions using this technique require the generation of special 2D space×time meshes that satisfy a cone (angle) constraint. The cone constraint requires that the dihedral angle of each interior mesh face with respect to the 2D space is less than or equal to a specified angle α. The value of α is determined by the fastest wave speed in the material (usually $\alpha = 45°$).

In recent years, there have been numerous advances in the generation of three-dimensional meshes. See [3] for an excellent review. Several methods have been suggested to address the specific problem of meshing the 2D space×time domain [4, 5]. However, none of those methods generates meshes that satisfy the cone constraint. Hence, new approaches to mesh generation, capable of handling the cone constraint, are required to use the DG analysis technique. Lowrie [6] presents a meshing scheme observing the cone constraint, but the scheme is applicable only when the initial quadrilateral mesh is structured and has uniformly sized elements.

The DG analysis puts few restrictions on the mesh element shapes, hence tetrahedral, hexahedral and possibly other element shapes are acceptable. In [7] we present a tetrahedral mesh generation scheme preserving the cone constraint. However, since hexahedral meshing schemes usually produce significantly fewer elements than tetrahedral ones, a hexahedral meshing scheme that satisfies the cone constraint is very desirable.

This paper presents a scheme that generates hexahedral meshes on space-time domains which satisfy the cone constraint. The algorithm builds the domain mesh as a sequence of layers in the time direction, starting from an initial quadrilateral mesh of the 2D space domain. It produces a high-quality hexahedral mesh that satisfies the cone constraint. The new method can adaptively adjust the geometry and the time-step size in response to the simulation.

The time-layers approach allows the method to handle geometry changes, as they occur, based on the simulation results, and to change the time-step size based on the analysis requirements.

Algorithm Overview

We consider the following constrained single layer meshing problem.

Single time slab problem: given a 2D unstructured quadrilateral mesh, construct a single layer of a space-time mesh such that, for each face of the mesh, the dihedral angle between it and the 2D space is less than or equal to α.

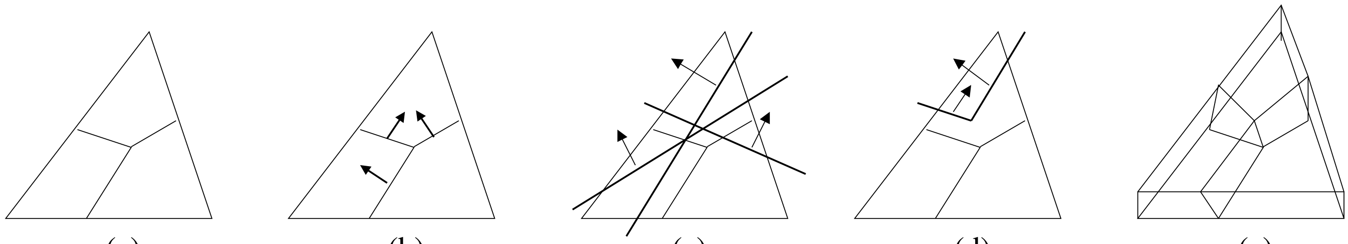
Once we have a solution for the single time slab problem, it is used to solve the general space-time mesh problem. The solution for each consecutive time slab is computed by inverting the mesh from the previous time slab and modifying it to conform to the new boundaries.

The space-time mesh layer is built by sweeping the 2D quadrilateral mesh in the time direction, by defining for each 2D mesh edge a sweep face which maintains the cone constraint. We describe below the algorithm for creating a consistent set of such edge faces, while maintaining mesh quality and conformity at common vertices.

The input for the algorithm is a 2D quadrilateral mesh Q_0 of the initial space domain. The mesh can be constructed using any available quadrilateral meshing technique. For the space-time meshing algorithm described here it is assumed that all interior nodes in Q_0 have valence three, four or five. This assumption is not restrictive since, as shown in [8], nodes with other valences can be replaced by a set of nodes with three, four or five valence through a set of topological modifications of the quadrilateral mesh. We also assume that at interior nodes of valence n the angles between adjacent edges are near $\frac{2\pi}{n}$. This condition can be achieved by using mesh smoothing algorithms, such as [9].

The first mesh layer is constructed by defining for each mesh edge in Q_0 a sweep face in the time direction which maintains the cone constraint. The construction algorithms is:

- For each interior edge e of Q_0, define a *right* or *left* tilting direction which is consistent with the directions defined for the neighboring edges.
- Based on the time slab height h, compute the necessary horizontal shift s for the edges. E.g., the horizontal shift will be similar to the height for a 45° cone constraint.
- For each edge e, compute the line e_t for the tilted edge which is parallel to and at a distance s from the edge. According to the tilting direction the line is to the left or right of the edge.
- For each node n, compute a new node n_h, that is at distance h from n in the time direction. Compute the 2D space position of n_h based on the tilted edges e_t attached to it and the half spaces defined by them.
- Define the new mesh connectivity based on the constructed new nodes. Define the quadrilateral mesh Q_1 from the new nodes n_h.



(a) (b) (c) (d) (e)

Figure 1: Stages of a mesh layer construction: (a) initial quad mesh; (b) tilting directions set for the edges; (c) the tilted edge lines e_t and the legal half-spaces defined by them (shown by the arrows); (d) the half-space intersection - legal area for node placement; (e) final mesh, based on setting the new node at the half-space intersection corner.

Figure 1 illustrates these steps for a simple example. The following two sections explain the steps in detail.

After a single mesh layer H_0 is computed, each next layer H_i is created by mirroring the previous layer H_{i-1} about the Q_i plane and modifying it to fit the domain boundaries (Figure 6(b)). Q_i is defined from the new mesh nodes at time-slab distance h from Q_{i-1}.

The procedure produces a hexahedral mesh of the space-time domain that satisfies the cone constraint.

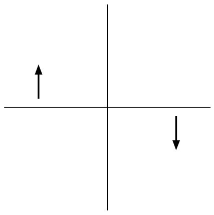
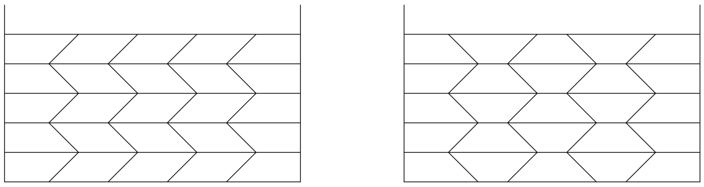


Figure 2: Inconsistent tilting of edges. Tilting the edges in the same direction (relative to the common node) results in an empty half-space intersection for the node position; i.e., there is no position for the node which satisfies both edge directions.

Setting Edge Tilting Directions

To define the sweep faces of the mesh in the time direction, a tilting direction needs to be defined for each interior mesh edge e. An edge can tilt either left or right. However, since a direction for the edge defines the tilting direction for the edges' end nodes, the directions of all the edges that share a common node have to be consistent. Figure 2 shows an inconsistent edge setting. Hence, an algorithm which sets the edge tilting directions has to guarantee a consistent setting.

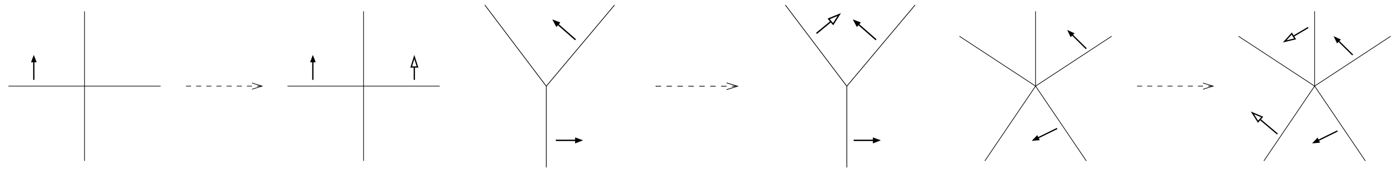


(a) (b)

Figure 3: Direction setting within a layer: (a) single direction; (b) opposite directions for opposite edges/faces.

Another consideration in the design of a tilting algorithm is that the shape of the mesh can influence the analysis results. If the mesh faces in each layer are tilted in a uniform direction (Figure 3(a)), than this imposes an artificial direction on the analysis and may affect its results. To avoid this, the tilting of the edges has to be non-uniform. Optimally, a tilting algorithm should provide opposite tilting directions for each pair of opposite faces of a hexahedron (Figure 3(b)).

The algorithm in Figure 5 was developed based on these two considerations. As shown in Figure 4, setting the direction of some of the edges at a node often forces a specific direction to be set at other edges to avoid inconsistencies. In the implemented algorithm, we perform a check at every node to see if such a constrained setting should be forced, and, if so, follow it.



(a) (b) (c)

Figure 4: Constrained edge direction setting for different node valences: (a) four valence node (one out of two constrained setting cases); (b) three valence node (the only constrained setting case); (c) five valence node (one out of ten constrained setting cases).

Definitions:
$dir \in (left/right/none)$ - edge tilt direction
n - node
$e = (e_s, e_e)$ - edge with nodes e_s, e_e (interior edges only)
d_e - tilt direction of e
S - Stack of tuples of $(node, dir)$

Functions:
$boolean\ contradiction(n)$ - check if n has contradiction in edge setting
$dir\ constrainedDir(n, e)$ - check if the current edge setting at n sets a direction for e and return this direction

Algorithm:
1 $dir\ nodeDir = right$
2 Going along the region boundary for each node n
3 $S \leftarrow (n, nodeDir)$
4 reverse $nodeDir$
5 call **EmptyStack**
6 while $\exists e = (e_s, e_e)$ s.t. $d_e = none$
7 d_e =random$(left, right)$
8 $S \leftarrow (e_s, none)$
9 $S \leftarrow (e_e, none)$
10 call **EmptyStack**

EmptyStack
1 while $S \neq \emptyset$
2 $S \rightarrow (n, dir_n)$
3 if $contradiction(n)$
4 ERROR
5 for each $e = (n, m)$ at n
6 if $dir_e \neq none$
7 continue (at 5)
8 $cdir = constrainedDir(n, e)$
9 if $cdir \neq none$
10 $dir_e = cdir$
11 $S \leftarrow (m, none)$
12 elseif $dir_n \neq none$
13 $dir_e = dir_n$
14 $S \leftarrow (m, none)$

Figure 5: Edge direction setting algorithm.

Using this method it is provable that, for a mesh with only four and three-valence nodes there will be no contradictions. From our experience, for practical purposes, this is true for five-valence nodes as well.

Node Placement

After the directions for the edges are set, the node positions for the new mesh layer can be computed. Based on the cone constraint angle α and the height h of the time slab (h is related to both analysis needs and the quadrilateral mesh Q element size), each new edge e' has to be offset from the original edge e at least a distance s (in the 2D space), where s is defined by $s = h/\tan(\alpha)$.

To satisfy this constraint for each edge e, a line e_t is computed such that e_t is parallel to e; according to the tilting direction, e_t is to the left or right of e; and the distance from e to e_t is s. The half-space hs_{e_t} defined by e_t (the half not containing e) is an acceptable space for e' nodes. Figure 1(c) shows an example of the half-spaces for a set of edges at a three-valence node.

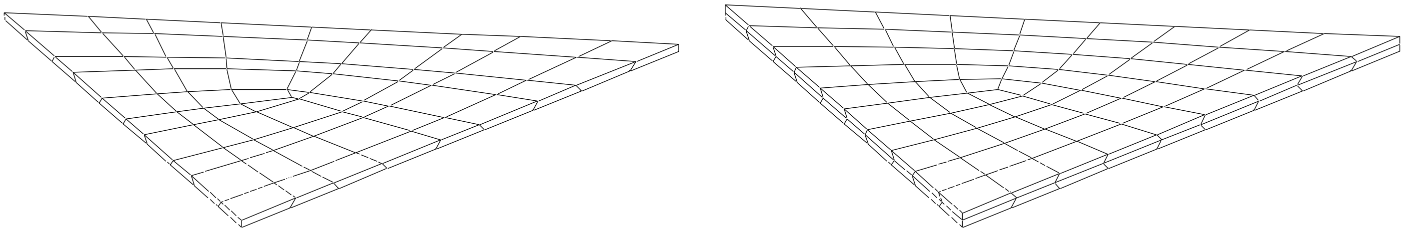
Thus for each node n, the acceptable space for placement of the new node n' is the intersection of the half-spaces defined by the edges starting at the node. In Figure 1(d), a region formed at a three-valence node is shown. The nearest position to the original node is chosen within this region. The region is guaranteed to be non-empty, since the edge directions are consistent.

The new mesh node n' is placed at the chosen position in the 2D space, and at a distance h from the original node in the time direction.

After defining the positions of nodes in the new layer, a layer of hexahedral mesh elements can be constructed. This provides a starting layer for the mesh. From here on, each additional layer is constructed by tilting the previous layer as explained above.

Results

The algorithm has been implemented and tested successfully on several geometries. We show three example geometries and the resulting meshes in Figures 6 and 7.



(a) (b)

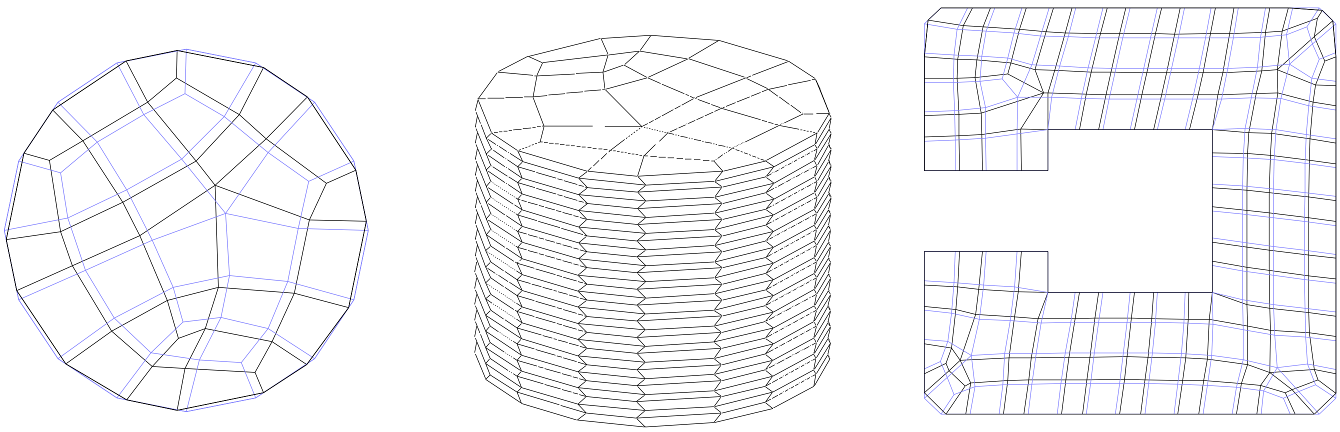
Figure 6: Mesh of a triangular region: (a) first mesh layer; (b) second layer created by inverting the first.

Conclusions

Discontinuous Galerkin methods provide the means to perform direct element-by-element solutions of complex second-order hyperbolic problems. This eliminates the need to solve large global systems of equations. However a new approach for space-time meshing is required to use these methods, so as to ensure satisfaction of the cone constraint.

A new algorithm for constructing hexahedral meshes which satisfy the cone constraint is presented in this paper. The method generates high-quality hexahedral meshes of the space-time domain, based on an initial quadrilateral mesh of the 2D space and a description of the space-time boundary. The method builds the mesh by layers (time-slabs). This allows it to handle geometry changes of the domain as determined by the simulation, and to adjust the time-step size based on the analysis requirements.

The main goal for future research is extension of the algorithm to 3D$\times$time meshes. Other areas of research include improvement of the edge direction choice technique to make it more efficient, mesh smoothing and handling of topological changes in the mesh boundary (such as the introduction of new boundary loops).



(a) (b) (c)

Figure 7: (a,b) mesh of a cylinder (a) the edge tilting (2D view); (b) domain mesh; (c) mesh of a "C" shape: the edge tilting (2D view). In the 2D views, the darker (black) lines show the tilted edges.

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).

References

1. L. Yin, A. Acharya, N. Sobh, R.B. Haber, and D. A. Tortorelli. A space-time discontinuous galerkin method for elastodynamic analysis. *Proc. of Int. Symp. on Discontinuous Galerkin Methods, Salve Regina University, Newport, RI*, 1999.
2. M. E. Gurtin. *The Linear Theory of Elasticity, Handbuch der Physik 6a_2*. Springer-Verlag, Berlin, 1972.
3. S. J. Owen. A survey of unstructured mesh generation technology. *7th International Meshing Roundtable, Sandia National Laboratories*, SAND98-2250:239–267, 1998.
4. P. J. Zwart and G. D. Raithby. Space-time meshing for two-dimensional moving boundary problems. *7th International Meshing Roundtable, Sandia National Laboratories*, SAND98-2250:187–200, 1998.
5. A. M. Froncioni, A. Garon, and R. Camarero. h-adaptive strategies for space-time finite elements. *Proc. 3rd Anual Conference of the CFD Society of Canada*, pages 47–54, 1995.
6. R. B. Lowrie, P. L. Roe, and B. van Leer. Space-time methods for hyperbolic conservation laws. In *Barriers and Challenges in Computational Fluid Dynamics*, volume 6 of *ICASE/LaRC Interdisciplinary Series in Science and Engineering*, pages 79–98. Kluwer, 1998.
7. A. Üngör, C. Heeren, X.Y. Li, A. Sheffer, R. B. Haber, and S. H. Teng. Constrained 2D space-time meshing with all tetrahedra. *To be presented at 16th IMACS World Congress*, 2000.
8. M. L. Staten and S. A. Canann. Post refinement element shape improvement for quadrilateral meshes. *Trends in Unstructured Mesh Generation, ASME*, AMD-Vol. 220:9–16, 1997.
9. P. M. Knupp. Winslow smoothing on two-dimensional unstructured meshes. *7th International Meshing Roundtable, Sandia National Laboratories*, SAND98-2250:449–458, 1998.