SAND2008-3853C

A Scheme for Automatic, Local, Conformal Refinement of Hexahedral Meshes*†

Jackson R. Mayo, David Thompson, and Philippe Pébay

Sandia National Laboratories
P.O. Box 969, Livermore CA 94551, U.S.A.
{jmayo,dcthomp,pppebay}@sandia.gov

Summary. This paper presents a new method for partitioning an input polytope into a conformal set of polytopes according to a subdivision pattern specified on lower-dimensional boundaries of the input polytope. A feature of this method is that it is applicable to any dimension. In three dimensions, for input polytopes that are hexahedral, all members of the partition will be hexahedral as well. A further feature of this method is that it preserves convexity; a convex input polytope will result in a conformal partition composed of convex polytopes. Finally, the method may be used to process large conformal meshes of input polytopes in parallel and produce a globally conformal partition without any communication between processes.

1 Introduction

Some applications of the finite element method – such as solid mechanics where shear of thin walls is involved – perform better with an all-hexahedral mesh than with a tetrahedral mesh [1] by virtue of requiring fewer degrees of freedom. However, all-hexahedral meshes can be difficult to use because

- most hexahedral mesh generators require user intervention, or fail with some inputs, or require a significant amount of processing time compared to tetrahedral techniques;
- current techniques for the h-refinement of hexahedral meshes require communication of refinement patterns to neighboring elements and iteration until all elements have a pattern that can be accommodated, or produce meshes with pendant nodes; and
- it is more difficult to detect and correct hexahedra with poor geometric quality that can cause simulations to diverge.

*The authors have submitted a provisional patent application on this work.

†This work was supported by the United States Department of Energy, Office of Defense Programs. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

As problem sizes grow, these tasks must be performed in parallel – which adds a stiff penalty to methods requiring communication of boundary information between portions of a model residing in different processes. What is needed is some local, conformal refinement scheme. We have developed such a scheme using homothetic transforms to partition a hexahedron into subhexahedra according to some indicator function. The next section reviews homothetic transforms, complexes, and indicator functions. These concepts allow us to introduce a simple scheme for local refinement that always produces convex subhexahedra given a convex input; for the sake of generality, we describe the scheme for polytopes in an arbitrary dimension. Then we explore some rules to produce fewer and better-quality subhexahedra. Finally, we discuss the implications for parallel, all-hexahedral refinement and mesh generation.

2 Mathematical Background

Before describing the scheme for local, conformal refinement of polytopes, a review of some terminology from three areas of mathematics is necessary: homothetic transforms, complexes, and indicator functions. All our definitions are made in a fixed spatial dimension d.

A *homothetic transform* (or *homothecy*) is a transformation $H\colon \mathbb{R}^d \to \mathbb{R}^d$ consisting solely of translation and uniform scaling. It can thus be written

$$H(\mathbf{x}) = \mathbf{o} + c(\mathbf{x} - \mathbf{o}), \tag{1}$$

where $\mathbf{o} \in \mathbb{R}^d$ is the *center* of the scaling operation and $c \in \mathbb{R}$ is the *contraction* or *dilation factor*. We are interested in cases where $c < 1$ so that points are moved closer to one another and $c > 0$ to avoid a singularity or inversion. These constraints on c mean that H will always perform a contraction. In particular, $H(\mathbf{x})$ lies on the line segment between $\mathbf{o}$ and $\mathbf{x}$, and thus a convex set $S \subseteq \mathbb{R}^d$ containing $\mathbf{o}$ will have $H(S) \subseteq S$.

Now we turn to the definition of a complex, the mathematical representation of a conformal mesh of convex polytopes. A *k-plane* ($0 \le k \le d$) is a set

$$\{\, \mathbf{a} + \mathbf{x} \mid \mathbf{x} \in V \,\}, \tag{2}$$

where $\mathbf{a} \in \mathbb{R}^d$ and the vector space V is a k-dimensional subspace of $\mathbb{R}^d$. Additionally, we define the only (-1)-*plane* to be the empty set $\emptyset$. Note that a 0-plane consists of a single point, and the only d-plane is $\mathbb{R}^d$. A *k-region* ($0 \le k \le d$) is a subset of a k-plane that is not a subset of any $(k-1)$-plane. Because the intersection of two different k-planes is a subset of a $(k-1)$-plane, the k-plane associated with a k-region is unique. A *k-face* is a k-region that is compact and convex. Terms such as the boundary and interior of a k-face are defined with respect to the standard topology of its enclosing k-plane (rather than that of $\mathbb{R}^d$). Note that a 1-face is a line segment.

Before defining a complex, we note some properties of faces related to contractive homothetic transforms. The image of a k-face f under such a transform H is a (smaller) k-face. Further, for a contraction factor c and a center $\mathbf{o}$ not in the k-plane of f, we define the *sweep* of f under H as

$$\bigcup_{c \le b \le 1} \{\, \mathbf{o} + b(\mathbf{x} - \mathbf{o}) \mid \mathbf{x} \in f \,\}. \tag{3}$$

The sweep is a $(k+1)$-region because the parameter b introduces an additional dimension, and it is compact and convex because f has these properties. Thus the sweep of a k-face is a $(k+1)$-face. We will momentarily be interested in k-faces whose boundaries are unions of $(k-1)$-faces. Note that the sweep of such a k-face f has for its boundary the union of the following k-faces: f, the image of f, and the sweeps of the $(k-1)$-faces forming the boundary of f.

A *complex* is a finite set C of faces such that, for $1 \le k \le d$,

- the boundary of every k-face in C is a union of $(k-1)$-faces in C, each enclosed in a different $(k-1)$-plane,
- every $(k-1)$-face in C is a subset of the boundary of a k-face in C, and
- the intersection of any two different k-faces in C is either $\emptyset$ or a j-face in C with $j < k$.

A (convex) *polytope* can be formally defined as a complex containing exactly one d-face (although the term is sometimes used for the d-face itself). Every complex is a union of polytopes associated with each of its d-faces.

An *indicator function* (or *characteristic function*) is an injective map χ from some set of interest S to the set $\{0, 1\}$. In our application, the domain of this map will be the set of m-faces (m fixed, $1 \le m \le d-1$) of a d-dimensional input complex. Elements of S which χ maps to 1 will be considered "marked" for subdivision while elements of S which χ maps to 0 must remain whole as the input complex is partitioned into output primitives.

An *implicit function* f is a function which is not defined explicitly, but rather in terms of a relationship, for example of the form $f(\mathbf{x}) = 0$. Under relatively weak conditions, the locus of points that satisfy the function form a $(d-1)$-dimensional manifold embedded in $\mathbb{R}^d$. Where *explicit functions* provide a parametric enumeration of the members of some manifold embedded in $\mathbb{R}^d$ (i.e., an expression $\mathbf{x}(\mathbf{t})$ which lists all $\mathbf{x}$ for which $f(\mathbf{x}) = 0$ with $\mathbf{t}$ in some simple domain), implicit functions must be inverted to identify members of the manifold where the equation holds. Implicit functions, and specifically those which estimate the geometric signed distance to some manifold, are commonly used in computer graphics as indicator functions that identify the boundary of a surface to be rendered. Our application will use implicit indicator functions to identify faces of polytopes where refinement should occur. If some point $\mathbf{x}$ on an m-face of an input polytope satisfies $|f(\mathbf{x})| < \epsilon$, the face will be considered "marked" for subdivision. An indicator function of the form

$$\chi(s) = \begin{cases} 1 & \dfrac{|f(\mathbf{x}(s))|}{\epsilon(s)} < 1, \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

may then be used to identify polytope faces that are near the surface of some implicitly defined level set. Note that where f is smooth, $\boldsymbol{\nabla} f(\mathbf{x})$ is defined and is a vector pointing away from the interior of the region bounded by $f = 0$.

3 Refinement Method

3.1 Basic Scheme

The goal of our method can be stated as follows: Given a complex C, an integer m with $1 \leq m \leq d - 1$, and a “refinement” set $R \subseteq C$ of m-faces selected by an indicator function, produce a new complex C' consisting of at least

- all k-faces in C with $k < m$,
- all m-faces in $C \backslash R$, and
- for each m-face $f \in R$, a plurality of m-faces whose union is f.

Our method will have the useful feature that, if every d-face in C is isomorphic to a d-cube, then every d-face in C' is isomorphic to a d-cube. The method will be stated and justified generally, and then stated concretely for the case believed to be most useful in practice, with $d = 3$ and $m = 1$, corresponding to edge-based refinement of hexahedral meshes. The meaning of the conditions above is then that the edges in R are refined (split into more than one edge) while the remaining edges are not modified, and other changes are made as necessary to preserve conformality.

To describe the method, we define a *k-target* ($k \geq m$) as a k-face in the initial complex C that is a superset of a face in R. Starting with $C_{-1} \equiv C$, the method applies the following operations sequentially for $\ell = 0, 1, \ldots, d - m$, producing a sequence of complexes $C_0, C_1, \ldots, C_{d-m} \equiv C'$.

Iterate over every $(d-\ell)$-target f. Choose a contractive homothetic transform H with center in the interior of f. Define a *k-superface* of f as a k-face in $C_{\ell-1}$ that is a superset of f. To construct C_ℓ, the superfaces of f are removed from $C_{\ell-1}$ and replaced with

- the images under H of all faces $h \in C_{\ell-1}$ that are subsets of d-superfaces of f and
- the sweeps under H of all faces $h \in C_{\ell-1}$ that are subsets of d-superfaces of f and whose planes are disjoint from the interior of f.

The indicator function can play a role in choosing the center and contraction factor of H.

Because the homothetic transforms are contractive and always shrink faces, a face arising at any stage as an image cannot be be a superset of a face in C, and one arising as a sweep of a face h is a superset of a face $g \in C$ if and only if $h \supseteq g$. Also, if every d-face in C is isomorphic to a d-cube, then every $(d-1)$-face in C is isomorphic to a $(d-1)$-cube. The new d-faces in C_0 arise as images of d-faces in C and as sweeps of $(d-1)$-faces in C; both are isomorphic to d-cubes. Continuing the process, it follows that every d-face in $C_{d-m} \equiv C'$ is isomorphic to a d-cube.

There is no ambiguity as to which $(d-\ell)$-target is involved in removing a given face, because of a key separation property: Each face in $C_{\ell-1}$ is a superset of at most one $(d-\ell)$-target. This is true for $\ell = 0$ because each face in C is a superset of at most one d-face in C. Assuming that the separation property is true up to but not including a given $\ell \geq 1$, so that $C_{\ell-1}$ can be constructed, note that every face $g \in C_{\ell-2}$ that is a superset of a $(d-\ell)$-target is either a k-target with $k \geq d-\ell$ or a sweep, sweep of a sweep, ... of such a k-target. If $k = d-\ell$, then g is a superset of exactly one $(d-\ell)$-target; if $k \geq d-\ell+1$, then g is a superset of a $(d-\ell+1)$-target and thus g is removed in constructing $C_{\ell-1}$. Hence each face in $C_{\ell-2}$ that is retained in $C_{\ell-1}$ is a superset of at most one $(d-\ell)$-target.

Every new face added in $C_{\ell-1}$ is either an image (which cannot be a superset of a target), or a sweep of a face $h \in C_{\ell-2}$, where h is a subset of a d-face $p \in C_{\ell-2}$ that is, by hypothesis, a superset of exactly one $(d-\ell+1)$-target f. Such a sweep is a superset of a $(d-\ell)$-target only if h is either a k-target with $k \geq d-\ell$ or a sweep, sweep of a sweep, ... of such a k-target. If $k \geq d-\ell+1$, however, then h would be a superset of a $(d-\ell+1)$-target, which, being a subset of p, must be f; but $h \supseteq f$ contradicts the condition that the plane of h is disjoint from the interior of f. Therefore $k = d-\ell$, and the sweep of h is a superset of exactly one $(d-\ell)$-target. The separation property then follows.

To see that each of the constructions $C_0, \ldots, C_{d-m} \equiv C'$ is actually a complex, note that when a face is removed, it is replaced by faces with consistent internal boundaries, due to the matching between images and sweeps. The external bounding faces are either unchanged or replaced by unique templates arising from a chosen homothetic transform for a given k-target. Thus conformality is preserved. The final complex C' satisfies the conditions stated at the beginning of this section. Only k-faces with $k \geq m$ are ever removed, and so C' contains all k-faces in C with $k < m$. Also, m-faces are removed only in the last stage, with $\ell = d-m$, when each m-target f (i.e., each face $f \in R$) is removed. Among the added faces are a plurality of m-faces whose union is f: the image of f, and the sweeps of the $(m-1)$-faces forming the boundary of f, under a contractive homothetic transform with center in the interior of f. No m-face in $C \backslash R$ is removed.

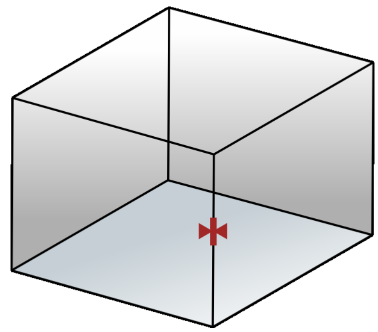

Fig. 1. A single input hexahedron with one edge marked.

3.2 Hexahedra

As a concrete example of the basic scheme, consider the case where $d = 3$, our complex C is composed solely of convex hexahedra (isomorphic to 3-cubes), and $m = 1$, i.e., χ is defined on edges (1-faces). An example would be the hexahedron of Figure 1 with one edge marked.

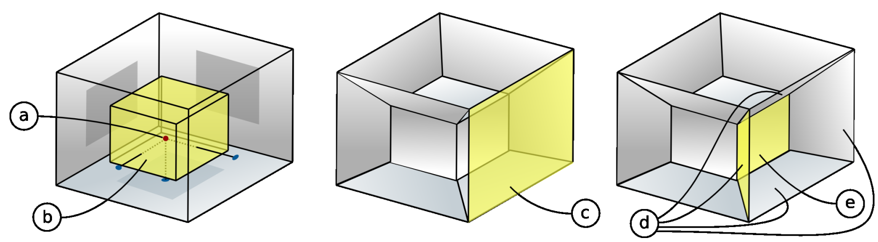

Fig. 2. A homothetic transform with its center (a) interior to the convex input hexahedron. The input hexahedron and its contracted image (b) have corresponding 2-faces (c) and (e) respectively, which can be connected along corresponding edges by 2-faces (d) to partition the input hexahedron into a set of convex hexahedra. The six outer hexahedra are sweeps of the 2-faces of the input hexahedron. Some 2-faces are not shown and yellow 2-faces are semi-transparent in order to illustrate the geometry more clearly.

Whenever any edge (1-face) of an input hexahedron is marked, the hexahedron is a 3-target and we first ($\ell = 0$) use an interior homothecy to partition the input hexahedron into seven smaller hexahedra, as shown in Figure 2. Of these seven hexahedra, six (the sweeps) share a 2-face with the input hexahedron. The separation property is evident, in that each of the new hexahedra contains at most one 2-target (input 2-face with one or more marked edges). Each hexahedron containing a 2-target will have a 2-face-bound homothecy applied ($\ell = 1$), followed by an edge-bound homothecy applied to any resulting hexahedron bounded by a marked edge of the input hexahedron ($\ell = 2$).

The results of 2-face-bound and edge-bound homothetic transforms applied to a rectangular prism are shown in Figure 3. Note that the hexahedra to which we apply homothecies are not generally prismatic, but the figures are much easier to comprehend when illustrated with prismatic hexahedra.

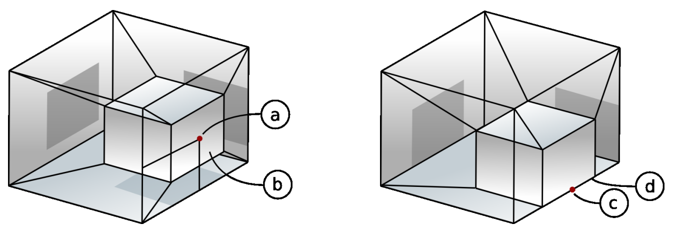

Fig. 3. Homothetic transforms with their centers – (a) and (c) – interior to a 2-face or edge (1-face) of the input hexahedron, leaving a 2-face (b) or edge (d) of the contracted hexahedron coincident with the input hexahedron.

To finish the example, the 2-face-bound and edge-bound homothecies required to subdivide the input hexahedron of Figure 1 on its single marked edge are shown in Figure 4.

3.3 Modified Scheme for Fewer, Better-Quality Hexahedra

Although the scheme described so far is simple and general, not all the homothetic transforms carried out are always necessary. The three-step process for edge-based hexahedral refinement ($\ell = 0, 1, 2$) is designed to enforce the separation property at each step, but if this property happens to hold already, one or more homothecies can be skipped without violating mesh conformality.

Specifically, if all marked edges of the input hexahedron border a single 2-face, then the interior homothecy is not needed; a 2-face-bound homothecy can be applied directly to the input hexahedron, as in Figure 3 (left). Each resulting hexahedron with a marked edge is then subdivided using edge-bound homothecies. Also, if exactly one edge of the input hexahedron is marked (as in the example above), then neither interior nor 2-face-bound homothecies are actually needed; an edge-bound homothecy can be applied directly to the input hexahedron, as in Figure 3 (right). Finally, if an interior homothecy is needed but a particular input 2-face has exactly one marked edge, then the hexahedron arising as the sweep of that 2-face does not require a 2-face-bound homothecy to isolate the edges from one another; an edge-bound homothecy can be applied directly to this hexahedron. If these special rules are followed, conformality is still automatic as long as the same homothecy is used consistently for all hexahedra sharing a given 2-face or edge.

Both the original and modified schemes result in a final subdivision with a maximum of 133 hexahedra, attained when all 12 edges of the input hexahedron are marked. But, over the $2^{12} = 4096$ possible sets of marked edges, the

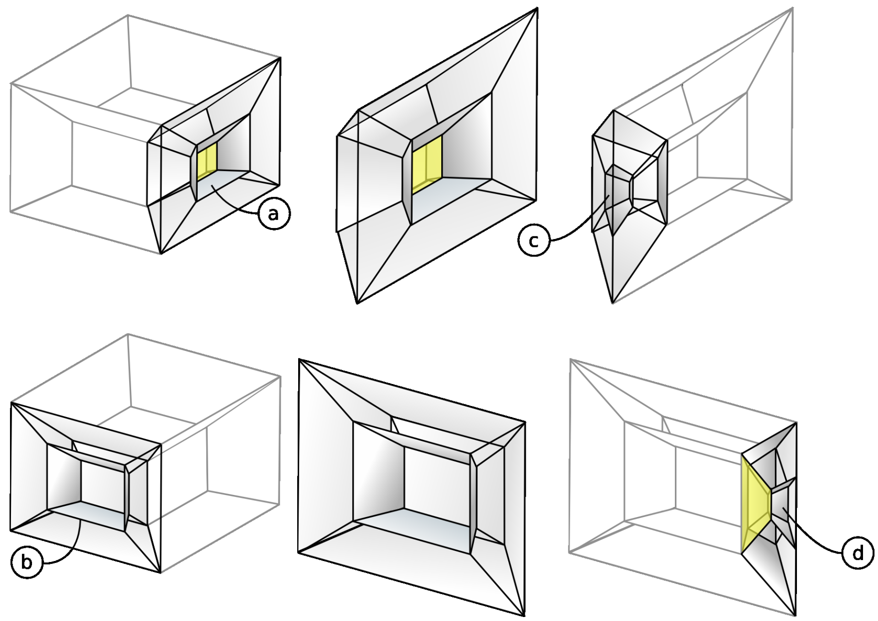

Fig. 4. This sequence starts after an initial interior homothecy has been applied. It shows how two 2-face-centered and two edge-centered homothecies can be applied to refine the edge specified in Figure 1. Left column: Homothecies are applied with centers on 2-faces (a) and (b), since they border an edge marked for subdivision. Middle column: A larger version of the 2-face-centered homothecies. Right column: Hexahedra bounded by the indicated edge are transformed to obtain edge-contractions (c) and (d).

original scheme produces an average of 83.12 hexahedra, while the modified scheme produces an average of 75.33 hexahedra. Not only does the modified scheme give a more computationally efficient mesh, but because each successive subdivision somewhat degrades the geometric quality of hexahedra, the omission of some homothecies is expected to result in a better quality of hexahedra in the refined mesh.

4 Future Work

While the algorithm for refining a conformal mesh of input polytopes according to a specified set of m-face subdivisions has been discussed, its use in a large application has not been covered. Because the indicator functions on each face of an input complex may be independently specified, and because the replacement rules are local to a given d-face and its subfaces, the refinement of polytopes may take place concurrently without any communication, as long as each process makes an identical choice of homothetic transform for a given shared target. If vertices induced by subdivision need to be shared by multiple output polytopes, communication of unique identifiers for these vertices is required, but this may be performed in a single stage after refinement has occurred. This same approach has worked for simplicial mesh refinement [8].

Another topic we have only briefly mentioned is the choice of indicator function. We are aware of simplicial refinement techniques that use the chord distance from an edge midpoint $\mathbf{x}_m$ to the level set of an implicit, i.e.,

$$\inf \left\{ |\mathbf{x}_m - \mathbf{x}| \;\middle|\; \phi(\mathbf{x}) = 0 \right\} < \epsilon; \tag{5}$$

the difference in the direction of the gradient vector of an implicit between edge endpoints $\mathbf{x}_0$ and $\mathbf{x}_1$, i.e.,

$$\boldsymbol{\nabla}\phi(\mathbf{x}_0) \cdot \boldsymbol{\nabla}\phi(\mathbf{x}_1) < (1 - \epsilon)\, |\boldsymbol{\nabla}\phi(\mathbf{x}_0)|\, |\boldsymbol{\nabla}\phi(\mathbf{x}_1)|; \tag{6}$$

or the value of an implicit function evaluated at an edge midpoint, i.e.,

$$\phi(\mathbf{x}_m) < \epsilon. \tag{7}$$

But we are not aware of any that compare the direction of $\boldsymbol{\nabla}\phi$ to the direction of the edge. An expression which compares these two directions in an indicator function might be used to favor subdivision of edges normal (or tangential) to a surface of interest. For the hexahedral case, this may tend to produce boundary constraints that are more favorable to high-quality output when the gradient does not vary significantly over input hexahedra; edges normal to some level set of ϕ would be refined more than other edges producing a layer of hexahedra near the level set surface. If refinement was also more likely on edges tangent to the gradient, the surface itself would be sampled more finely. Edges not aligned with the gradient would leave larger hexahedra in transition regions. We expect that the ratio of the edge length to the gradient magnitude may be a good normalization factor for indicator functions of this type.

References

1. Steven E. Benzley, Ernest Perry, Karl Merkley, Brett Clark, and Greg Sjaardema. A comparison of all hexagonal and all tetrahedral finite element meshes for elastic and elasto-plastic analysis. In *Proceedings of the 4^{th} International Meshing Roundtable*, Oct. 16–17 1995.
2. Carlos D. Carbonera and Jason F. Shepherd. A constructive approach to constrained hexahedral mesh generation. In *Proceedings of the 15^{th} International Meshing Roundtable*, pages 435–452. Springer-Verlag, Sep. 17–20 2006.
3. Ichiro Nishigaki, Masayuki Hariya, and Yoshimitsu Hiro. 6,578,189, June 2003. Hexahedral mesh generation method and device.
4. Philippe P. Pébay and David Thompson. Communication-free streaming mesh refinement. *Journal of Computing and Information Science in Engineering*, 5 (4):309–316, 2005. Special Issue on Mesh-based Geometry.

5. M. Staten, S. Owen, and T. Blacker. Unconstrained paving and plastering: A new idea for all-hexahedral mesh generation. In *Proceedings of the 14th International Meshing Roundtable*, pages 399–416. Springer-Verlag, Sep. 11–14 2005.
6. Matthew L. Staten, Robert A. Kerr, Steven J. Owen, and Ted D. Blacker. Unconstrained paving and plastering: Progress update. In *Proceedings of the 15th International Meshing Roundtable*, pages 469–486. Springer-Verlag, Sep. 17–20 2006.
7. Timothy J. Tautges and Sarah E. Knoop. Combinatorial aspects of dual-based hexahedral mesh modification. In *SIAM Workshop on Combinatorial Scientific Computing*, Feb. 27–28 2004.
8. David Thompson and Philippe P. Pébay. Embarrassingly parallel streaming mesh refinement. *Engineering With Computers*, 22(2):75–93, August 2006.
9. Phillip Tuchinsky and Brett W. Clark. The hex-tet, hex-dominant automesher: An interim progress report. In *Proceedings of the 6th International Meshing Roundtable*, Oct. 13–15 1997.
10. Yongjie Zhang and Chandrajit Bajaj. Adaptive and quality quadrilateral/hexahedral meshing from volumetric data. In *Proceedings of the 13th International Meshing Roundtable*, Sep. 19–22 2004.