Control Volume Meshes using Sphere Packing: Generation, Refinement and Coarsening

Gary L. Miller* Dafna Talmor* Shang-Hua Teng† Noel Walkington‡ Han Wang‡

Abstract. In this paper we present a sphere-packing technique for Delaunay-based mesh generation, refinement and coarsening. We have previously established [10] that a bounded radius of ratio of circumscribed sphere to smallest tetrahedra edge is sufficient to get optimal rates of convergence for approximate solutions of Poisson's equation constructed using control volume (CVM) techniques. This translates to Delaunay meshes whose dual, the Voronoi cells diagram, is well-shaped. These meshes are easier to generate in 3D than finite element meshes, as they allow for an element called a *sliver*.

We first support our previous results by providing experimental evidence of the robustness of the CVM over a mesh with slivers. We then outline a simple and efficient sphere packing technique to generate a 3D boundary conforming Delaunay-based mesh. We also apply our sphere-packing technique to the problem of automatic mesh coarsening. As an added benefit, we obtain a simple 2D mesh coarsening algorithm that is optimal for finite element meshes as well.

1 Introduction

The control volume method (CVM) is a popular method for solving partial differential equations, especially when the underlying physical problem has some conservation properties, such as for heat or flow problems. The domain is partitioned into small "control" volumes, and the approximation is derived using the conservation properties over each control volume. The Voronoi diagram, and its dual, the Delaunay triangulation, are particularly fitted for the control volume method, with the Voronoi cells acting as the control volumes, and the Delaunay triangulation as the neighborhood structure of interacting Voronoi cells.

The quality of a triangular or simplicial mesh is measured in terms of both the shape of its elements and their number. Unstructured Delaunay triangulation meshes are often used in conjunction with the finite element method (FEM). In the context of the finite element meshes, the element shape quality is measured by its aspect ratio. Several definitions for the element aspect ratio exist, and they are all equivalent up to a constant factor: the ratio of the element height to the length of the base, the ratio of the largest inscribed sphere to the smallest circumscribed sphere, or the size of the smallest element angle.

The problem of generating good aspect ratio meshes has been discussed extensively in the literature (see the survey by Bern and Eppstein [2]). CVM mesh generation was assumed to pose the same requirements as FEM mesh generation, and did not merit its own discussion. However, we have shown recently [10] that the

*School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213. Supported in part by NSF Grant CCR-9016641

†Department of Computer Science, University of Minnesota, Minneapolis, Minnesota 55455. Supported in part by an NSF CAREER award (CCR-9502540). Part of this work was done at Department of Mathematics and the Lab. for Computer Science, MIT, Cambridge, MA.

‡Department of Mathematics, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213.

error estimates of the CVM depend on a shape criteria weaker than the aspect-ratio. In particular, CVM meshes require elements with a bounded ratio of radius of circumscribed sphere to smallest element edge, among other requirements mentioned below. In 2D, this *radius-edge ratio* is equivalent to the standard aspect ratio, but in 3D it allows a very flat element with arbitrarily small dihedral angles, and edges proportional to the circumscribed sphere radius (see Figure 1). This type of element is referred to as a *sliver*.

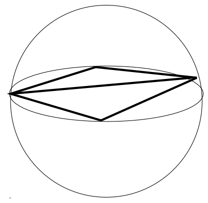

Figure 1: Sliver: a very flat tetrahedra, with good radius-edge ratio

Generating a mesh of good radius-edge ratio appropriate for the CVM is the motivation behind this paper. In 3D, it is an easier task than generating a mesh with a good aspect ratio. Our approach is to use an approximate sphere-packing technique. In this paper we analyze the conditions under which a sphere-packing yields a good radius-edge mesh that conforms to a 3D domain description.

These are what we consider to be the main contributions of this paper:

- Comparing and contrasting the FEM and the CVM over a mesh with slivers. We provide experimental evidence that the CVM is insensitive to slivers.
- Characterizing the requirements of CVM mesh generation.
- Proposing a new sphere-packing approach: we analyze the conditions under which a sphere-packing yields a good radius-edge mesh that conforms to a 3D domain description. This also provides some insight into the workings of other related mesh generation methods, such as advancing front methods and Ruppert's method, and also into the development of efficient 3D mesh generation.
- Applying the new sphere-packing technique to 2D provably optimal automatic mesh coarsening. In 2D, this automatic mesh coarsening applies to good aspect-ratio (FEM) meshes as well.

1.1 Previous Work

The study of mesh generation abounds in the literature. Bern, Eppstein and Gilbert [4] had the first provably optimal 2D good aspect-ratio mesh generation algorithm. The mesh they generated is based on a quad-tree. Ruppert [15] presented a simpler algorithm that produces smaller meshes in practice. Ruppert's algorithm is based on a Delaunay triangulation, and extends the approach Chew [6] suggested for quasi-uniform meshes to general unstructured meshes.

3D mesh generation is still a mostly uncharted research area for good aspect-ratio meshes. Mitchell and Vavasis [11] extended to 3D the quad-tree approach of Bern, Eppstein and Gilbert [4]. The simpler Delaunay triangulation approach of Ruppert and Chew has not been successfully generalized. The work of Dey, Bajaj and Sugihara [7] is a generalization of Chew's algorithm into 3D. It is targeted at quasi-uniform meshes only, and addresses only simple convex boundaries. Even for that simpler case their 3D generalization of Chew's approach fails to generate good aspect ratio Delaunay triangulation. It does, however, generate good bounded radius-edge ratio meshes.

The failure of the Dey, Bajaj and Sugihara algorithm to produce good aspect-ratio meshes preempted any discussion on how to generalize Ruppert's related algorithm to 3D. Now that the weaker condition of radius-edge ratio is proven useful, the question of generalization of Ruppert's algorithm regains its relevance. Even if this generalization were straight-forward, it would merit a lengthy discussion so that the issues of conforming to 3D boundaries are thoroughly investigated.

However, the obvious generalization of Ruppert's algorithm to 3D suffers from some drawbacks. Recall that the basis of Ruppert's scheme is to iteratively produce the target point set by maintaining a Delaunay triangulation of the point set, picking any bad aspect-ratio triangle, adding its circumcenter to the point set or splitting a segment if appropriate, retriangulating the point set and repeating until all triangles are well-shaped. This scheme is simple and efficient in 2D in practice, though is not theoretically guaranteed to be efficient. In 3D, two facts hinder us: (1) The intermediate 3D triangulation maintained is potentially of size $O(|P|^2)$, P being the intermediate point set, whereas in 2D the triangulation is always linear. (2) In 2D the triangulation is repaired by an edge flipping algorithm; the complexity of a repair can be $O(n)$ but in practice is smaller. In 3D each edge flip is more complex, and furthermore the total number of flips necessary to update the Delaunay diagram can be of a worst case $O(n^2)$. (Note however that in this case, as the starting point of the incremental construction is a DT, the edge flips algorithm is guaranteed to converge to a Delaunay diagram. This has been shown separately by Joe [8], and Rajan [14]).

A different technique is therefore necessary to produce an efficient generalization of Ruppert's algorithm to 3D. In this paper we concentrate on one such technique: approximate sphere-packing. It has been observed before [16] that a pattern of tightly packed spheres led in nature to well shaped Delaunay triangulations and Voronoi diagrams.

We are aware of at least two instances where sphere packing was used as a basis for mesh generation, but with a different goal and underlying technique: (1) Shimada's bubble packing: spheres are packed in the domain in some initial configuration, and then a physically based iterative method is used to smooth the mesh and obtain a better shaped mesh. Our method is not an iterative method, and its emphasis is on laying down the spheres such that the corresponding Delaunay triangulation carries radius-edge ratio guarantees. As such, it is orthogonal to Shimada's method. (2) Bern, Mitchell and Ruppert [3] used sphere packing to generate an $O(n)$ non-obtuse triangulation of a polygonal domain in 2D. The elements produced in that instance could be of bad aspect ratio and radius-edge ratio.

The rest of this paper is as follows: Section 2 provides definitions and experimental results related to the CVM method. Section 3 outlines the sphere-packing technique for mesh generation and refinement, and Section 4 presents the automatic mesh coarsening.

2 The Control Volume Method and Meshes

In this section we review the control volume method (CVM). The CVM associates to each interior mesh point a volume. These volumes are non-overlapping regions whose union approximates the domain. The Voronoi diagram provides a natural choice of control volumes. In the next subsections we define the Voronoi diagram and its dual, the Delaunay triangulation.[1] We recall our previous error estimate [10] for the CVM and the mesh structure this error estimate requires, and present a numerical experiment comparing and contrasting the finite element and control volume algorithms on such meshes.

2.1 Voronoi Diagrams and Delaunay Triangulations

Given a set of points $P = \{p_1, \ldots, p_n\} \subset \mathbb{R}^d$, the convex hull of $d+1$ linearly independent points from P forms a *Delaunay simplex* if their circumscribed ball contains no point from P in its interior. The union of all Delaunay simplices forms the *Delaunay triangulation*, $DT(P)$. If the set P is not degenerate then $DT(P)$ is a simplicial decomposition of the convex hull of P. We refer to the circumsphere of a simplex in $DT(P)$ as a Delaunay ball.

The geometric dual of the Delaunay Diagram is the *Voronoi Diagram*, and consists of a set of *Voronoi Polyhedra* $\{V_1, \ldots, V_n\}$, one for each point in P. Geometrically, V_i is the set of points $p \in \mathbb{R}^d$ whose Euclidean distance to p_i is less than or equal to that of any other point in P. We call p_i the *center* of the polyhedra V_i. The Delaunay triangulation is dual in the sense that two points p_i, p_j form an edge in $DT(P)$ if and only if V_i and V_j share a common face. In this instance, the Delaunay edge is perpendicular to the common face.

[1] We adopt the standard terminology where a decomposition of a region in an $\mathbb{R}^d$ into simplices is referred to as a triangulation.

2.2 Control Volume Discretizations

We now review the control volume technique for approximating Poisson's equation:

$$-\Delta u = f, \qquad \text{in } \Omega, \qquad\qquad u|_{\partial\Omega} = g,$$

where $\Omega \subset \mathbb{R}^d$ is a bounded domain, and Δ is the Laplacian (e.g. in three dimensions, $\Delta u = u_{xx} + u_{yy} + u_{zz}$). For simplicity we assume that Ω is a polygonal domain so that it can be triangulated exactly, and we consider Delaunay triangulations of Ω whose Voronoi regions corresponding to interior vertices are contained within Ω. Letting V_i be the Voronoi region corresponding to an interior vertex p_i, we integrate the equation for u over V_i to get

$$\int_{V_i} f = \int_{V_i} -\Delta u = \int_{\partial V_i} -\frac{\partial u}{\partial n},$$

where the second equality follows upon integration by parts, and ∂V_i refers to the boundary of V_i. Let the length of the Delaunay edge joining vertex p_i to p_j be denoted by h_{ij}, and let $\mathcal{N}_i$ to be the set of indices j such that p_j is connected to p_i by an edge, i.e. the set of Delaunay neighbors of p_i. For each Delaunay edge there is an associated Voronoi face (or edge in two dimensions) common to V_i and V_j, which we denote by A_{ij} (see Figure 2). Letting u_i be an approximation of $u(p_i)$ (u being the exact solution), the above equation is approximated by:

$$|V_i| f_i = \int_{\partial V_i} -\frac{\partial u}{\partial n} = \sum_{j \in \mathcal{N}_i} \int_{A_{ij}} -\frac{\partial u}{\partial n} \simeq \sum_{j \in \mathcal{N}_i} |A_{ij}| \frac{u_i - u_j}{h_{ij}}.$$

In the above, f_i is the average value of f over the Voronoi V_i. This equation is to hold for each interior vertex p_i, and on the boundary we set $u_i = g(p_i)$. The solution of the resulting linear system gives the control volume approximation of Poisson's equation. It is transparent that the matrix corresponding to this system of linear equations is an M-matrix, so that the discrete maximum principle holds. MacNeal [9] shows that in two dimensions this matrix is identical to that given by the finite element method constructed using piecewise linear functions on the Delaunay triangulation; however, this is not the case in three dimensions.

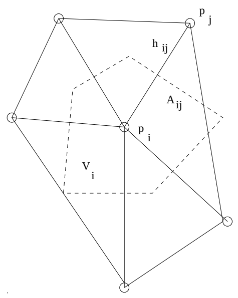

Figure 2: The Voronoi diagram of p_i, the Delaunay edge h_{ij}, and the Voronoi face A_{ij}.

2.3 Error Estimates and Mesh Geometry

We recall the error estimates Nicolaides [12] established for the control volume method. For quantities defined on the edges of the Delaunay mesh define the inner product $(.,.)_W$ by

$$(U, V)_W = \sum_{(i,j)} |A_{ij}| h_{ij} U_{ij} V_{ij}$$

and corresponding norm $||.||_W$ as

$$||U||_W^2 = (U, U)_W,$$

where $\sum_{(i,j)}$ indicates summation over all of the non-boundary Delaunay edges.

Theorem 2.1 (Nicolaides) *Let $\{u_i\}$ be the discrete solution given by the control volume method, and define $U_{ij} = (u_i - u_j)/h_{ij}$ and $U_{ij}^{(1)}$ and $U_{ij}^{(2)}$ by*

$$U_{ij}^{(1)} = \frac{u(p_i) - u(p_j)}{h_{ij}} \qquad U_{ij}^{(2)} = \frac{1}{|A_{ij}|} \int_{A_{ij}} -\frac{\partial u}{\partial n}$$

(u is the exact solution) then $||U - U^{(1)}||_W \leq ||U^{(1)} - U^{(2)}||_W$.

Note that the right hand side of this error estimate depends only upon the exact solution, and $||U||_W$ is a discrete version of the $L^2(\Omega)$ norm of the gradient.

Nicolaides proceeded to estimate the error $||U^{(1)} - U^{(2)}||_W$ under the assumption that the meshes had simplices with bounded ratio of circum to inscribed sphere radius ratio; however, we showed previously [10] that this hypothesis can be relaxed to the situation where the *radius-edge* aspect ratio is bounded.

Definition 2.2 *The* radius-edge *ratio of a simplex is the ratio of the radius of the circumscribed sphere to the shortest edge of the simplex.*

Theorem 2.3 *Let $\rho = \max_{(i,j)} r_{ij}/h_{ij}$ be the radius-edge aspect ratio of a Delaunay triangulation of Ω, then*

$$||U^{(1)} - U^{(2)}||_W \leq (1 + 4\rho^2)||D^2 u||_{L^2(\Omega)} h,$$

where $D^2 u$ is the Hessian (matrix of second derivatives) of u, and $h = \max_{(i,j)} h_{ij}$.

Note that, as usual, the estimate above can be localized in the sense that the right hand side is actually the sum of products of h_{ij} and the L^2 norm of the second derivatives in small regions containing the edge. For this reason it is natural to refine the mesh in regions where the second derivatives are large.

2.4 Control Volume Meshes

The previous section established that a bounded radius-edge ratio is sufficient for optimal rates of convergence for approximate solutions of Poisson's equation constructed using the control volume method. Classical finite element theory relies on a different aspect ratio condition: the smallest angle in the simplex is required to be larger than some fixed constant. In 2D, the bounded smallest angle condition and the bounded radius-edge ratio condition are equivalent up to a constant factor. In 3D, the radius-edge ratio is weaker than the smallest angle bound. A *sliver* is an element whose radius-aspect ratio is bounded by a constant, but whose smallest (dihedral) angle can be as small as desired (see Figure 1). A CVM mesh relies on a weaker condition, and consequently the generation of good 3D CVM meshes is an easier task.

As the dual Voronoi diagram plays a major role in the CVM, it is of interest to understand the bounded radius-edge aspect ratio in the dual setting. The Voronoi cells of points of P that are on the convex hull of P are infinite cells. Internal Voronoi cells are finite. In the next theorem we refer to a clipped form of the infinite Voronoi cell (clipped by the boundary or the convex hull); the proof appeared previously [10].

Theorem 2.4 *Let P be a set of points in $\mathbb{R}^d$, C a constant such that the radius-edge ratio of $DT(P)$ is bounded by C. Let R be the radius of the smallest ball containing the (clipped or finite) Voronoi cell $V(p)$, r the radius of the largest ball contained in $V(p)$. There exists a constant D depending on C and the dimension d only, such that $R/r \leq D$.*

Intuitively, Theorem 2.4 implies the Voronoi cells resemble a sphere. These well-shaped Voronoi cells are at the heart of the optimal rate of convergence of the CVM over bounded radius-edge ratio Delaunay based meshes.

The physical meaning and the mathematical treatment of the CVM is more natural if the Voronoi cells of interior mesh points are contained in the domain. It is therefore natural to consider algorithms that generate such meshes. More accurately, if B_i is some boundary feature (an input point, segment or face) we require that $p \notin B_i \Rightarrow V(p) \cap B_i = \emptyset$.

h	ϵ	d/ρ	FEM $\|\|u-u_h\|\|_{H^1}$	FEM $\|\|I_h u - u_h\|\|_W$	CVM $\|\|I_h u - u_h\|\|_W$
1/4	0.01	1126	30.22479	7.696399	0.2425438
	0.001	11125	92.97309	8.692323	0.2411516
	0.0001	111296	291.6406	8.917005	0.2410718
1/8	0.01	26230	12.98964	4.344101	0.07617430
	0.001	166051	35.13775	6.865007	0.07531520
	0.0001	1621745	108.1435	7.525118	0.07538803
1/16	0.01	14194	6.023042	2.262189	0.02155074
	0.001	105551	15.48025	4.457009	0.02005826
	0.0001	1030956	45.91799	6.140328	0.02007369

Figure 3: Comparison of finite element and control volume methods.

2.5 Numerical Experiments

To illustrate the effectiveness of the CVM on meshes with slivers we consider approximating the function u defined on the unit cube given by

$$u(x,y,z) = e^{\pi x}\cos(\pi y/\sqrt{2})\sin(\pi z/\sqrt{2})$$

which satisfies $-\Delta u = 0$. Meshes with slivers are constructed by randomly perturbing the points of a uniform mesh on the cube. The Delaunay triangulation of the perturbed points contains slivers on the horizontal and vertical planes from which the points were perturbed. If the maximum perturbation is ϵ, the dihedral angle of the slivers decreases with ϵ.

Figure 2.5 compares the control volume and finite element solutions for this problem. The radius-edge ratio for all of the meshes considered is of order one, unlike the maximum ratio of circum to inscribed sphere radius that is shown for each mesh. The difference between the interpolated exact solution ($I_h u$) and the numerical solution for each approximation is tabulated. Classical finite element theory estimates the error in the H_0^1 norm[2], and this is included for the finite element solution.

It is clear from Figure 2.5 that the error in the control volume approximations is independent of ϵ, as predicted by the theorem. Also, the finite element solution deteriorates as the slivers become flatter and flatter. Rates of convergence can also be estimated from the table. Note that for $\epsilon = 0.01$ the error in the finite element solution reduces by a factor of two when the mesh size is similarly reduced, implying unit rate. While Theorem 2.3 also predicts a first order rate of convergence for the control volume approximations, a better rate is achieved for this example; in fact the error almost reduces by a factor of four as the mesh size is reduced by one half, indicating a second order rate of convergence. This is due to the extra symmetry of the uniform mesh. On a uniform mesh the control volume scheme is second order accurate; indeed, it is clear from the proof of Theorem 2.3 that the error is proportional to h^2 when all the Delaunay edges intersect their corresponding Voronoi faces at their centroid.

Finite element theory shows that the finite element solution has the smallest error in the $\mathcal{H}_0^1$ of any piecewise linear function on a particular mesh. In particular, this error is smaller than that obtained by interpolating the exact solution. The interpolant of the exact solution will have small H_0^1 errors on tetrahedra having large ratios of circumscribed to inscribed sphere radius, and it is the slivers where the errors in the gradient are large. It was observed that the control volume solution behaved in a similar fashion to the interpolatant. For example, if the H_0^1 norm is computed on the domain formed by removing the slivers (which typically occupy less than 1% of the volume), then both the interpolant and control volume solutions gave errors smaller than those of the finite element solution which spreads the error more uniformly among the tetrahedra.

[2] $\|u\|^2_{H_0^1(\Omega)} = \int_\Omega |\nabla u|^2$

3 Mesh Generation using Sphere-Packing

In this section we introduce the notion of a sphere-packing and show how it can be used to generate boundary conforming Delaunay based meshes with bounded aspect ratio and optimal size up to a constant factor. This approach has a large overlap in ideas with that of Ruppert [15], but yet quite different conceptually. It can be thought of as a framework for generating Delaunay-based meshes, which generalizes Ruppert's approach.

Let D be some domain in $\mathbb{R}^d$ and f a function from D to the positive reals. We say that f is *α-Lipschitz* if $|f(x) - f(y)| \leq \alpha \|x - y\|$ for all points x, y in the domain. In this case we call f a *spacing function* for D with Lipschitz constant α. An *f-sphere* of a point $p \in D$ is a sphere with center p and radius $f(p)$.

A set of points $\mathcal{P} = \{p_1, \cdots, p_n\} \subset D$ is a *sphere-packing* with respect to a spacing function f if the interiors of their f-spheres do not intersect, i.e., $|f(p_i) + f(p_j)| \leq \|p_i - p_j\|$ for $i \neq j$. It is *maximal* if $\mathcal{P}$ is a maximal such set. We later consider relaxed forms of the notion of maximality, and the different mesh generation algorithms they suggest.

As a simple example of the kind of results one can prove for sphere-packings, let D be all of $\mathbb{R}^d$ and f be the constant function $f(x) = 1$. A packing $\mathcal{P}$ is then just a packing of unit radii spheres. If $\mathcal{P}$ is maximal then the Delaunay Diagram of $\mathcal{P}$ has radius-edge ratio at most one; by the fact that it is a packing every edge must have length at least 2. The radius of every Delaunay ball must be at most two, otherwise the center of the ball can be added to the packing, contradicting maximality. More generally one can show the following:

Lemma 3.1 *If $D = \mathbb{R}^d$, f is α-Lipschitz, and $\mathcal{P}$ is a maximal sphere-packing of D then the Delaunay Diagram of $\mathcal{P}$ have radius-edge ratio at most $1/(1 - 3\alpha)$.*

Proof: Let e be the shortest edge of some Delaunay simplex whose circumball has center c and radius r. Further let p be the endpoint of e that minimizes f. Since it is a packing, $2f(p) \leq |e|$. Since $\mathcal{P}$ is maximal, there exists a point $q \in \mathcal{P}$ that prevents c from being added, i.e., $f(c) + f(q) \geq \|c - q\| = r' \geq r$. Since f is α-Lipschitz, $f(q) \leq f(c) + \alpha r$. Using the last two facts, $2f(c) \geq r'(1 - \alpha) \geq r(1 - \alpha)$. Using again the Lipschitz property of f it is the case that $f(c) \leq f(p) + \alpha r$. Combining the previous facts, $2f(p) + 2\alpha r \geq r(1 - \alpha)$ or $|e| \geq 2f(p) \geq r(1 - 3\alpha)$. □

We next show how to use packing to generate Delaunay diagrams that conform to a boundary. The boundaries we consider are piecewise linear, a generalization of the notion of a planar straight line graph [2] to higher dimensions. A *polytope* is the convex combination of finite set of points P. The dimension of the polytope is the dimension of the affine hyper-plane generated by P. The boundaries as well as the domain we consider are a collection of polytopes. We formally use the following definition:

Definition 3.2 *A* piecewise linear system *is a set of polytopes S with the following properties:*

1. *The set $\mathcal{S}$ is closed under taking boundaries, i.e., for each $P \in \mathcal{S}$ the boundary of P is a union of polytopes in $\mathcal{S}$.*
2. *$\mathcal{S}$ is closed under intersection.*
3. *If $dim(P \cap Q) = dim(P)$ then $P \subseteq Q$, and $dim(P) < dim(Q)$.*

We also require that the angle between any two intersecting polytopes, when one is not contained in the other, is at least $90°$. The notion of a piecewise linear system is similar to a convex decomposition of non-manifold polyhedra [1], but is more general. For example, a box with a segment floating in its middle can be described as a piecewise linear system with no need to further decompose it.

A *k-simplex* S is a convex polytope of dimension k with $k+1$ vertices. The *d-circumball* of S is an open ball in d-dimensions, whose center and radius are equal to the center and radius of the ball circumscribed by S in the k-dimensional hyper-plane containing S.

To generate a Delaunay mesh conforming to the piecewise linear system description, the sphere-packing proceeds in increasing order of dimension, see Figure 4 for a pseudo-code description. First, we add the 0-polytopes to the packing, with their corresponding f-spheres. The spacing function f can be set to guarantee that the f-spheres do not intersect. Next, the 1-polytopes of $\mathcal{S}$, its segments, are sphere-packed: points are

Procedure: PACKING$(\mathcal{S}, f)$

Input: $\mathcal{S}$, a piecewise linear system of dimension 3.
f, a spacing function.

Output: A set of points P.

Method:

1. Let P_0 be of the 0-polytopes of $\mathcal{S}$. Let B_0 be an empty set of balls.
2. **For** $i = 1$ to 3
   (a) For each i-polytope $Q \in \mathcal{S}$ find a maximal sphere-packing P_i of Q with respect to f, under the following restrictions for each $p \in P_i \cap Q$:
      i. The f-sphere of p is disjoint from the f-sphere of any point $q \in \cup_{j=0}^{i} P_j \cap Q$.
      ii. p is disjoint from any open ball in $\cup_{j=0}^{i-1} B_j$ whose center is on Q.
      iii. For $i = 2$, p is far from each 1-polytope contained in Q (See Definition 3.3).
   (b) For each i-polytope $Q \in \mathcal{S}$ compute the Delaunay Diagram on Q using only the points of $(P_0 \cup \cdots \cup P_{i-1}) \cap Q$. For each i-simplex in that diagram find its d-circumball. Set B_i to be the collection of all these d-circumballs.

   end for
3. Return $P = P_0 \cup \cdots \cup P_3$.

Figure 4: 3-dimensional sphere-packing algorithm.

placed on the segments such that their f-spheres are disjoint, and disjoint from the f-spheres of the 0-polytopes. The packing of each segment naturally divides it into 1-simplices (edges). We want these edges to appear in the final Delaunay triangulation, and therefore an empty open ball must be maintained around each edge. We call these the edge *protecting balls*. The protective balls we use are the d-circumballs of the edges: balls centered at the middle of the edges, whose radius is half the edge length.

The 2-polytopes of $\mathcal{S}$ are then sphere-packed: a maximal set of points is placed on the faces such that their f-spheres are disjoint, and disjoint from previously placed f-spheres, and such that the points do not intersect the edge protecting balls. Each face is then Delaunay triangulated with respect to the points on that face, and the d-circumballs of these Delaunay triangles are used as protecting balls of the 2-polytopes.

In algorithm 4 we also maintain that points packed on a 2-polytope are far enough from the 1-polytopes contained in it. We now formalize this notion:

Definition 3.3 *Let Q be a 2-polytope. A 1-polytope Q_1 is contained in Q if $Q_1 \cap Q = Q_1$. We say $p \in Q$ is* far *from the 1-polytope Q_1 contained in Q if the $2f$-sphere of p (a sphere centred at p whose radius is twice the radius of the f-sphere of p) is disjoint from Q_1.*

Finally, the 3-polytopes of $\mathcal{S}$ are sphere-packed, by placing a maximal set of point such that their f-spheres are disjoint from each other and previously placed f-spheres, and such that the points are disjoint from the edge and face protective balls.

We call the output of the packing procedure a *maximal f-packing of $\mathcal{S}$*. The result of the packing should have three properties:

I. **Global packing:** In the procedure, every i-polytope was packed independently of other i-polytopes, and independently of all lower dimensional polytopes not strictly contained in it. Nonetheless, the result of the packing should be a sphere-packing globally: no two f-spheres in the packing should intersect. Step (a)-iii of the algorithm was added to guarantee that.

II. **Boundary conforming:** Step (a)-ii of the algorithm forced boundary simplices to be contained in empty open balls, to ensure they are also a simplex of the Delaunay triangulation. However, since we packed independently on disjoint polytopes we have to show the protective balls are indeed empty.

III. **Good quality mesh:** The Delaunay simplices should be of bounded radius-edge ratio, and the Voronoi cells of points in the interior of the d-polytopes should not intersect lower dimensional polytopes of $\mathcal{S}$. Step (a)-i of the algorithm is partly responsible for the bounded radius-edge ratio.

We first specify the function with respect to which we pack. Ruppert defined a function called the *local feature size*, denoted $\mathrm{lfs}(x)$. We extend lfs in the natural way to the piecewise linear system $\mathcal{S}$ by setting $\mathrm{lfs}(x)$ to equal the radius of a closed ball around x that intersects at least two disjoint polytopes of $\mathcal{S}$. Note that lfs is 1-Lipschitz. For our algorithm a suitable spacing function is $f = \alpha \mathrm{lfs}$ with some appropriate constant $\alpha < 1$. Fixing the parameter α in order to satisfy the requirements above is dimension dependent. In the discussion below we argue the three dimensional case.

The following Lemma states the global packing property. The proof, which we omit, relies on the $90°$ limit on the angle between polytopes, and the fact f-spheres on faces are placed far from the 1-polytopes.

Lemma 3.4 *Let P be the result of a 3-dimensional packing. If $\alpha \leq \sqrt{2} - 1$ then the f-spheres of points $p, q \in P$ laid independently on non-disjoint polytopes do not intersect.*

To prove boundary integrity, we have to show that protective balls in $B_1 \cup B_2$ contain no points of P in their interior. Protective balls on a polytope $S \in \mathcal{S}$ do not contain points from P placed on a S' intersecting S either by the $90°$ angle restriction between polytopes, or because of step (a)-ii of the algorithm. The next two Lemmas show that if α is small enough, a protective ball can not contain a point $p \in P$ laid on a disjoint polytope.

Lemma 3.5 *Let c be the center of the protective ball B, and let R its radius, then:*

1. *If $B \in B_1$ then $f(c) \geq R(1-\alpha)/2$*
2. *If $B \in B_2$ then $f(c) \geq R\frac{1-3\alpha}{2\sqrt{2}}$*

Proof: We show only the first part. Let a and b be the end points of the edge defining B, c is their midpoint. Assume, wlog, that $f(b) \geq f(a)$.

(1) maximality $\Rightarrow f(b) + f(c) > R$.

(2) Lipschitz condition $\Rightarrow f(b) \leq f(c) + \alpha R$.

(3) (1) and (2) $\Rightarrow f(c) \geq R(1-\alpha)/2$

□

Lemma 3.6 *Let B be a protective ball added when packing a polytope $S \in \mathcal{S}$. Let $T \in \mathcal{S}$ be a polytope disjoint from S, $p \in P \cap T$. For a small enough α, the open ball B and p are disjoint.*

Proof: Let R be B's radius, c its center. Let $d = ||c - p||$. We now fix the α for which $R \leq d$ and hence the ball and the point do not intersect. Lemma 3.5 showed $f(c) \geq R\delta$, where δ depends on α and on whether $B \in B_1$ or $B \in B_2$. Hence, $R \leq f(c)/\delta \leq \alpha \mathrm{lfs}(c)/\delta$. Since c and p lie on disjoint polytope, $\mathrm{lfs}(c) \leq d$ and $R \leq \alpha d/\delta$. Hence, it suffices to fix α such that $\alpha/\delta \leq 1$. For $B \in B_1$ this is true for $3\alpha \leq 1$. For $B \in B_2$ this is true for $(3 + 2\sqrt{2})\alpha \leq 1$. □

We now address the quality of the resulting Delaunay triangulation. The following is important for CVM meshes:

Lemma 3.7 *The Voronoi cell of a point $p \in P$ can intersect only a polytope $S \in \mathcal{S}$ that p belongs to.*

Proof: First, assume that p resides on a 3-polytope S. We now show that the Voronoi diagram of p intersects no 2-polytope contained in S, and hence can not intersect any 2-polytope. By way of contradiction, assume there exists a point q on some 2-polytope $S_2 \in \mathcal{S}$ contained in S that belongs to the Voronoi cell of p. According to our algorithm, S_2 was first triangulated, and protecting balls in B_2 were assigned to each Delaunay triangle. Hence, before p was added, q belonged to the Voronoi cell of some point $d \in S_2$. We show that if q was closer to p than to d, then p is inside a protective ball $B \in B_2$ which is a contradiction by the way we construct the packing. Divide the old Voronoi diagram of d into triangular sectors by connecting d to its Voronoi points, q is in one of the sectors. We look at the two triangle protecting balls associated with the two end points of this sector. If q was closer to p than to d, p must be in one of those balls. Similarly, p's Voronoi diagram can not touch lower level polytopes because of protecting balls in B_1. The case when p is on a 2-polytope or a 1-polytope can be argued similarly. □

Let T be a Delaunay tetrahedron generated by the algorithm. To show that the tetrahedron has bounded radius-edge ratio we consider the possible reasons why its circumcenter c can not be added to the maximal packing: (1) the f-sphere of c would intersect another f-sphere, (2) c belongs to a protecting ball, (3) c was not in the polytope under consideration. Lemma 3.7 rules that case out.

Case (1) was handled by Lemma 3.1, leaving only case (2), which the following Lemma addresses.

Lemma 3.8 *Let T be some Delaunay tetrahedron of a maximal packing, R the radius of T's circumball, c its center. Let d be the shortest edge of T. If c's f-sphere intersects some protective ball B, then:*

1. *If $B \in B_1$ then* $\frac{d}{R} \geq \frac{1-\alpha(3+2\sqrt{2})}{\sqrt{2}}$
2. *If $B \in B_2$ then* $\frac{d}{R} \geq \frac{1-\alpha(7+2\sqrt{2})}{2}$.

Proof: We show the first part only, the second part is proven similarly by substituting the appropriate bound from Lemma 3.5. Let q be the center of the protective ball B, Q its radius. Let L be the distance from c to q. Let $\delta_1 = (1-\alpha)/2$.

(1) c can not be added $\Rightarrow$ c is in the edge protecting ball and by Lemma 3.5 $f(c) \geq f(q) - \alpha Q \geq (\delta_1 - \alpha)Q$.

(2) c can not be added $\Rightarrow L \leq Q$.

(3) Triangle inequality $\Rightarrow L^2 + Q^2 \geq R^2 \Rightarrow Q \geq R/\sqrt{2}$.

(4) (1) and (3) $\Rightarrow f(c) \geq R(\delta_1 - \alpha)/\sqrt{2}$.

(5) $f(c) \leq f(p) + \alpha R \Rightarrow f(p) \geq R(\delta_1 - \alpha - \sqrt{2}\alpha)/\sqrt{2}$.

(6) $f(p)$ is one of the end points of the edge d. Wlog, we can assume it is the one for which $f(p) \leq d/2$.

□

There are several interesting issues with efficiently implementing our packing algorithm. The two main computational issues are (1) computing the spacing function f and (2) ensuring that the packing is maximal.

The spacing function can be approximated by using a quad-tree or oct-tree to compute the local feature size in linear work. The correctness of procedure PACKING is guaranteed for any α-Lipschitz function smaller than αlfs for an appropriately fixed α, thus an approximate spacing function suffices.

There are several ways to weaken the sphere-packing maximality condition, each of which suggests a different generation algorithm. We consider two ways. The first is the notion of *Voronoi maximality*. Observe that the lemmas above argued only the addition of center points of Delaunay circumballs, which are the Voronoi diagram points. We therefore define a sphere-packing to be Voronoi maximal if no circumcenter of a Delaunay ball can be added. This definition leads to a Ruppert style algorithm, where the sphere-packing is built iteratively. The centers of current Delaunay balls (at ascending order of dimension) are considered for addition to the sphere-packing. The algorithm terminates once no circumcenter can be added.

The other weakening of maximality we consider is approximate maximality. A sphere-packing is *ϵ-maximal* if each ϵf-ball contains a point that can not be added to the sphere packing, $\epsilon > 0$. The lemmas above can be restated and proved for ϵ-maximal packings.

Using ϵ-maximal packing allows us to analyze two new techniques for packing, oversampling and filtering. *Oversampling* samples a point from every ϵf-ball. *Filtering* removes a subset of the oversampling to obtain an approximate sphere-packing. Our filtering technique first builds a *conflict graph* CG on the sampled points. The graph CG has an edge between two sample points if their f-spheres intersect. We call taking a maximal independent set of CG filtering.

Oversampling and filtering can be extended to the boundary case in a natural way. In procedure PACKING we replace maximal packing with ϵ-packing. We use oversampling and filtering to guarantee that we generate an ϵ-packing at each level. We iteratively oversample and filter in ascending order of dimension, so that the conflict graph also contains edges corresponding to protective balls of lower dimensional objects. The next section discusses another application of this technique.

As mentioned above, a coarse oct-tree can be used as the basic data-structure to compute an approximation to lfs, and to find the CG edges. In the oversampling stage, each oct-tree box can be sampled uniformly. We have not yet evaluated the quality of the resulting mesh versus other 3-dimensional mesh generators.

4 Automatic mesh coarsening: sphere unpacking

The problem of automatic mesh coarsening is that of producing a hierarchy $M_1 \cdots M_k$ of respectively coarser good aspect ratio meshes given an initial fine unstructured mesh M_0. A coarsening M_{i+1} of mesh M_i is a mesh whose elements are point-wise larger than M_i's, but still conforms (possibly in a relaxed sense) to the same domain.

For meshes obtained in a refinement process, this hierarchy can be obtained by undoing the refinement process; this assumes intermediate meshes were kept. In contrast to that approach, automatic mesh coarsening assumes no knowledge of how the mesh was generated.

Such a hierarchy of coarsening meshes is necessary, for example, for the hierarchical and multi-level technique, which has become one of the most effective and successful numerical techniques for solving partial differential equations (PDEs). It is used in multi-grid methods [5] and multi-level domain decomposition. The use of unstructured meshes is inevitable in the solution of complex problems with more intricate domain geometry and solutions.

This section applies the sphere-packing technique of the previous section to the problem of two dimensional mesh coarsening. Recall that in two dimensions the radius-edge ratio is equivalent to the standard aspect ratio. Hence the 2D techniques of this section are relevant for FEM meshes as well as CVM meshes.

4.1 The Coarsening Hierarchy

We enumerate the assumptions we make about the initial unstructured mesh, and the requirements of the coarsening hierarchy:

- **The initial mesh** M_0: This is an unstructured, two dimensional bounded radius-edge ratio mesh. To simplify the discussion, this section concentrates on the important case of *quasi-uniform unstructured meshes*: a mesh whose element sizes differ only by a constant factor.

- **Nested Hierarchy**: Our algorithm generates a node-nested coarsening hierarchy. A coarsening hierarchy can be classified as *element-nested*, *node-nested* or *non-nested*. In general, a triangular unstructured mesh does not have any element-nested coarsening, unless it was carefully crafted as such.

- **The meshes** $M_0, \ldots, M_k$: The elements of two consecutive meshes in the hierarchy should approximate each other well; we require an element of mesh M_{i+1} to intersect at most a constant number of elements of M_i, for $0 \leq i < k$.

The problem of *automatic mesh coarsening* is that of producing a hierarchy with size and shape guarantees: $M_0 \cdots M_k$ must have a uniformly bounded element aspect-ratio, and the number of elements in each mesh M_{i+1} must be as small as possible while still conforming to the coarsening hierarchy restrictions.

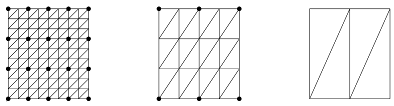

Figure 5: Repeated applications of MIS can degrade the aspect ratio. In the figure the ratio of the largest edge to the smallest edge is respectively 1, 3/2 and 9/4.

4.2 Previous Approaches

The problem of mesh coarsening has received much attention. Multi-grid papers either present a heuristic form of mesh coarsening algorithm, or assume the availability of a mesh coarsener with size and shape guarantees [13]. Outside of the Multi-grid domain, heuristic mesh coarsening techniques are used to reduce the size of meshes, as an intermediate step between the mesh generation and the solution phases [4].

The general method used for coarsening unstructured triangular meshes is that of picking some subset of the original mesh nodes and triangulating that set to form the coarsening. A commonly used technique is that of the Maximal Independent Set (MIS). The mesh is viewed as a graph, and a subset of the graph nodes is picked such that no two subset nodes share an edge (independence) and such that no node can be added to the set without violating independence (maximality).

The MIS technique is very successful in reducing the mesh size to a fraction of its original size; however it carries no guarantees for the other qualities of the mesh hierarchy, such as its aspect-ratio. The problem is illustrated in Figure 5: certain choices of an MIS of the original mesh degrade the aspect ratio of the coarser mesh. The aspect ratio degradation compounds with repeated applications. This can be observed even for highly uniform meshes.

We note that the MIS technique is usually employed randomly, and it would be intriguing to get probabilistic bounds for the aspect ratio of a mesh coarsening hierarchy thus generated; however the method outlined in the next subsection is simple, efficient and provides aspect-ratio and size guarantees.

4.3 Our Coarsening Algorithm

Repeated application of the MIS technique to construct a coarsening hierarchy can degrade the aspect ratio of the meshes. Intuitively, the information contained in each mesh is increasingly noisier. One way to avoid this escalating degradation is to build mesh M_i from mesh M_0 rather than from mesh M_{i-1}.

The method we propose uses the initial mesh M_0 to reconstruct a sphere-packing spacing function. A hierarchy of coarser spacing-functions is then used to guide the sphere-packing for each level of the coarsening hierarchy.

Refer to Figure 6 for the algorithm. We now discuss some of the steps:

- **Reconstructing the spacing function (step 1):** The first spacing function computed, f_0, is half the distance to the nearest neighbor. f_0 is 0.5-Lipschitz, since for any two points $p, q \in P_0$ $f_0(p) = r$ implies $f_0(q) \leq 0.5||p-q|| + r$. By the nearest neighbor definition the spheres of radii $\{f_0(p_i)|p_i \in P\}$ do not intersect, hence M_0 is a sphere-packing with respect to f_0.

- **The spacing function for mesh M_i (step $3a$):** f_i is generated by taking a maximum of a constant function and a 0.5-Lipschitz function and so is 0.5-Lipschitz itself. A simple max operation to obtain the spacing function of the coarser mesh is sufficient in that case since the original mesh was quasi-uniform. This has the effect of coarsening out only features smaller than a certain value. For more general meshes, it is desirable to coarsen out features of different sizes simultaneously.

  The size of the node set P_i associated with level i is small compared to $|P_0|$. It is inefficient to explicitly compute f_i for each and every point of P_0. Instead, the points P_0 are bucketed using buckets of size $O(2^i\mu)$. The function value $f_i(p)$ is explicitly computed only when point p is referred to. A variation of the algorithm can be implemented such that only a few points from each bucket are ever touched.

- **The node set of mesh M_i (steps $3a$-$3c$):** Increasing the spacing function causes a fraction of the points to be pruned out when the MIS of the conflict graph CG_i is generated. In step $3c$, Q can always

Algorithm: COARSEN(M_0)

Input: M_0, `a triangular mesh`

`Specified by:` P_0 `the mesh nodes;` E_0 `the mesh edges.`

k, `the number of levels in the output.`

Output: $M_1 \cdots M_k$, `the coarsening hierarchy meshes.`

Method:

1. $\forall p \in P_0$ `compute` $f_0(p) = 0.5 \min_{q \neq p; q \in P_0} ||p - q||$.
2. `Let` $\mu = \min_{P_0} f_0$.
   `Let` $Q = \emptyset$.
3. `For` $i = k$ `down to` 1
   - (a) $\forall p \in P_0$ `compute` $f_i(p) = \max(\mu 2^i, f_0(p))$.
   - (b) `Construct a conflict graph` $CG_i = (V, E)$ `of` P_0 `with respect to` f_i `as follows:` $V = P_0$ `and` $E = \{(p,q) : ||p - q|| < f_i(p) + f_i(q)\}$.
   - (c) `Let` P_i `be a completion of` Q `into an MIS of` CG_i.
   - (d) M_i `is a Delaunay triangulation of` P_i.
   - (e) $Q = P_i$.

   `end for`

Figure 6: The coarsening algorithm.

be completed to an MIS of CG_i since $CG_i \subseteq CG_{i+1}$. The conflict graph CG is an auxiliary graph, and is not generated explicitly in an efficient implementation of the algorithm. We use the bucketing scheme mentioned above. The MIS of CG is generated point by point. As a point is added to the MIS, all the points in buckets that conflict with it are removed from consideration.

The issue of the boundaries was omitted from the discussion so far. The geometry of the boundary poses restrictions on the amount of coarsening that can be done while still maintaining boundary integrity, or using good aspect ratio elements. The right way to coarsen the boundary beyond the boundary geometry restrictions is an interesting research issue. Our approach can be adapted to different solutions: (1) User supplied descriptions of the coarser boundaries or heuristically changing the boundary when coarsening beyond boundary integrity, and (2) Tolerating bad aspect ratio triangles on the boundary only.

Assuming coarsening beyond boundary restriction is not required; the coarsening algorithm 6 is adapted to the boundary case much like algorithm 4. The boundary is coarsened in ascending order of dimension. First, a conflict graph CG^0 with respect to the coarsening function is generated over the boundary input points, and a set of nodes S_0 which is an MIS of CG^0 is obtained. Then the conflict graph CG^1 is generated over the input segments. The node set of S_0 is completed to a node set S_1 which is an MIS of CG^1. The conflict graph CG^2 also contains the edge-spheres corresponding to the edges of nodes from S_1. S_2, the MIS of CG^2, is the final node set for the coarser mesh. Figure 4.3 demonstrates this approach, for the more interesting, non quasi-uniform, air-foil mesh. See Table 4.3 for the triangle statistics.

We sum up with the following theorem:

Theorem 4.1 *Let M_0 be a quasi-uniform mesh whose radius-edge ratio is bounded by ρ. Let $M_1 \cdots M_k$ be the coarsening hierarchy generated by algorithm* COARSEN. *There exist constants $\rho_1, \mathcal{I}$ and $C < 1$, depending on ρ only, such that:*

1. *The radius-edge ratio of all the elements of all the meshes $M_1 \cdots M_k$ is bounded by ρ_1.*
2. *For $i = 1 \cdots k-1$, $|M_{i+1}| < C|M_i|$, where $|M_{()}|$ is the number of nodes in the mesh.*

3. For $i = 1 \cdots k-1$, each element of mesh M_{i+1} intersects at most $\mathcal{I}$ elements of mesh M_i.

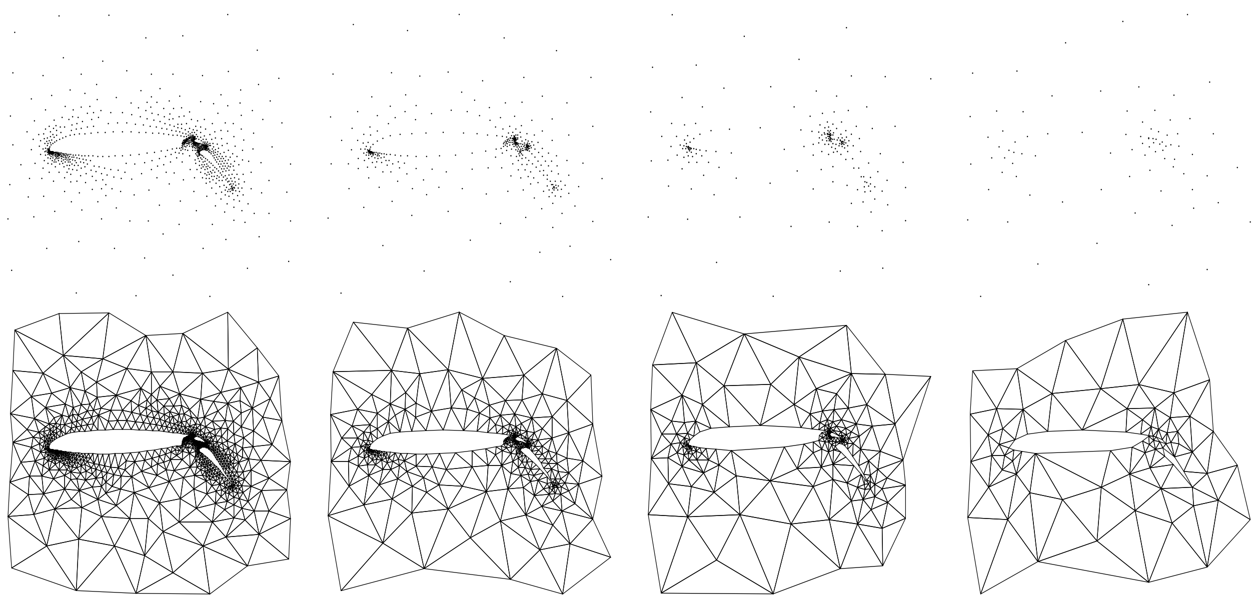

Figure 7: Repeatedly coarsening the air-foil mesh. No special attention was paid to coarsening the outside boundary, thus bad aspect ratio triangles formed by the coarsening on the outside boundary were removed. The internal boundaries were coarsened using sphere-packing, thus no bad aspect ratio triangles formed near them. To achieve that, the shape of the internal boundaries has to be coarsened as well.

number of nodes	number of elements	max angle	min angle	avg max angle	avg min angle
4253	8034	117.08	8.18	69.5	55.5
1039	1848	132.67	15.45	77.9	45.10
416	706	127.28	20.09	80.99	42.42
217	359	131.84	20.21	82.67	41.67
86	134	126.76	17.38	81.93	41.04

Figure 8: Angle statistics of the coarsened meshes. The first row presents statistics for the original mesh, the latter rows for the iterative coarsenings of the mesh.

References

[1] Chanderjit L. Bajaj and Tamal K. Dey. Convex decomposition of polyhedra and robustness. *SIAM J. Comput.*, 21(2):339–364, April 1992.

[2] M. Bern and D. Eppstein. Mesh generation and optimal triangulation. Tech. Report CSL-92-1, Xerox PARC, Palo Alto, CA, 1992.

[3] M. Bern, S. Mitchell, and J. Ruppert. Linear-size nonobtuse triangulation of polygons. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 221–230, 1994.

[4] Marshall Bern, David Eppstein, and John Gilbert. Provably good mesh generation. In *FOCS90*, pages 231–241, St. Louis, October 1990. IEEE.

[5] A. Brandt. Multi–level adaptive solutions to boundary value problems. *Mathematics of Computation*, 31:333–390, 1977.

[6] L.P. Chew. Guaranteed-quality triangular meshes. CS 89-983, Cornell, 1989.

[7] Tamal K. Dey, Chanderjit L. Bajaj, and Kokichi Sugihara. On good triangulations in three dimensions. In *Proceedings of the ACM symposium on solid modeling and CAD/CAM Applications 1991*, pages 431–441. ACM, 1991.

[8] B. Joe. Construction of three-dimensional Delaunay triangulations using local transformations. *Comput. Aided Geom. Design*, 8(2):123–142, May 1991.

[9] R. H. MacNeal. An asymmetrical finite difference network. *Quarterly of Applied Math*, 11:295–310, 1953.

[10] G. L. Miller, D. Talmor, S.-H. Teng, and N. Walkington. A Delaunay based numerical method for three dimensions: generation, formulation, and partition. In *Proc. 27th Annu. ACM Sympos. Theory Comput.*, pages 683–692, 1995.

[11] S. A. Mitchell and S. A. Vavasis. Quality mesh generation in three dimensions. In *Proceedings of the ACM Computational Geometry Conference*, pages 212–221, 1992. Also appeared as Cornell C.S. TR 92-1267.

[12] R. A. Nicolaides. Direct discretization of planar div–curl problems. *SIAM J. Numer. Anal.*, 29(1):32–56, 1992.

[13] Carl F. Ollivier-Gooch. Upwind acceleration of an upwind euler solver on unstructured meshes. *AIAA Journal*, 33(10):1822–1827, 1995.

[14] V. T. Rajan. Optimality of the Delaunay triangulation in R^d. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 357–363, 1991.

[15] Jim Ruppert. A new and simple algorithm for quality 2-dimensional mesh generation. In *Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 83–92, 1992.

[16] Kenji Shimada and David C. Gossard. Bubble mesh: Automated triangular meshing of non-manifold geometry by sphere packing. In *Proc. 3rd Sympos. on Solid Modeling and Applications.*, pages 409–419, 1995.