Well-Spaced Points for Numerical Methods

Dafna Talmor

August 1997

CMU-CS-97-164

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy*

Thesis Committee:
Gary L. Miller, *chair*
Guy Blelloch
Alan M. Frieze
Noel J. Walkington
Shang-Hua Teng, University of Minnesota

This research is sponsored by the National Science Foundation under Grant No. CCR-9505472. Views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Government.

Keywords: Computational geometry, mesh generation, mesh coarsening, multigrid

Abstract

A numerical method for the solution of a partial differential equation (PDE) requires the following steps: (1) discretizing the domain (mesh generation); (2) using an approximation method and the mesh to transform the problem into a linear system; (3) solving the linear system. The approximation error and convergence of the numerical method depend on the geometric quality of the mesh, which in turn depends on the size and shape of its elements. For example, the shape quality of a triangular mesh is measured by its element's aspect ratio.

In this work, we shift the focus to the geometric properties of the *nodes*, rather than the elements, of well shaped meshes. We introduce the concept of well-spaced points and their spacing functions, and show that these enable the development of simple and efficient algorithms for the different stages of the numerical solution of PDEs.

We first apply well-spaced point sets and their accompanying technology to *mesh coarsening*, a crucial step in the multigrid solution of a PDE. A good aspect-ratio coarsening sequence of an unstructured mesh $\mathcal{M}_0$ is a sequence of good aspect-ratio meshes $\mathcal{M}_1, \ldots, \mathcal{M}_k$ such that $\mathcal{M}_i$ is an approximation of $\mathcal{M}_{i-1}$ containing fewer nodes and elements. We present a new approach to coarsening that guarantees the sequence is also of optimal size and width up to a constant factor — the first coarsening method that provides these guarantees. We also present experimental results, based on an implementation of our approach, that substantiate the theoretical claims.

In three dimensions, we apply well-spaced points to mesh generation. We introduce a new aspect-ratio condition, the radius-edge ratio, which corresponds to well-spaced points. Radius-edge ratio is weaker than the standard aspect-ratio condition in that it allows slivers. Nonetheless, we show that a Delaunay mesh of bounded radius-edge ratio has geometric structure properties that are reminiscent of a bounded standard aspect-ratio mesh: the mesh is a density graph and its Voronoi cells are well-shaped. We finally show how to construct such meshes to conform to an input domain.

Acknowledgements

In the course of preparing this work I had the help, support and collaboration of many people, but foremost among them are Gary Miller and Shang-Hua Teng. I am deeply grateful for their close help and colaboration. Gary served as my advisor, guided me in my research path, and never ceased to astonish me with fresh view points and ideas and his enthusiasm for tackling new problems. Shang-Hua served both as a fellow student and as a research collaborator and mentor. His creativity and joy at solving problems made working with him fun and exciting. He was the first to suggest that the techniques of well-spaced points and spacing functions developed together with Gary might be applicable to the problem of mesh coarsening, and was instrumental in showing that that was indeed the case. I am also grateful to my other committee members, Guy Blelloch, Alan Frieze and Noel Walkington for their help and guidance at different points in my thesis.

I am thankful to Paul Heckbert, who introduced me to Poisson darts and their use in graphics, gave me many useful pointers and advice, and was always interested in, and supportive of, my work. Poisson darts, together with Jim Ruppert's mesh generation ideas, inspired the concept of well-spaced points. Thanks also to Michael Garland for sharing his and Paul Heckbert's simplification work with me, providing demos and patiently answering my questions.

For many inspiring discussions, I am grateful to Steve Guattery, Keith Gremban and Jonathan Shewchuk. My research crossed theirs many times, leading to wonderful and exciting glimpses of how my tiny research domain relates to the bigger picture. In Jonathan's case, our research areas became congruent rather than just crossing. I am grateful to Jonathan for many hours of discussions, for loads of useful pointers and papers, for his program `triangle` which I used extensively in this thesis and embedded in my mesh coarsener, and for the program `showme` which was used to generate many of the figures in this document. Also, I am grateful for the comments he gave on my thesis draft. No one talks about mesh generation with quite that twinkle in his eyes!

The Scandal group served as my informal project group. Thanks to all the Scandal group members: Margaret Reid-Miller, Girija Narlikar, Jay Sipelstein, Marco Zagha, John Greiner, Jonathan Hardwick and Guy Blelloch for all the support and advice, and the many many practice talks they suffered through. Guy Blelloch was also inspirational in my work on designing and implementing a parallel Delaunay triangulation algorithm in NESL, together with Gary Miller. I learned a lot about the Delaunay triangulation and implementation and evaluation issues during this work. Thanks also to Jonathan Hardwick who later took this experimental evaluation and made it a reality.

Dennis Grinberg, Jennie Kay and Anja Feldmann generously sat at some of my practice talks, and were very supportive at those phases of the program I did not find entirely enjoyable. Their encouragement was crucial to me, and I will miss them greatly. I am also grateful to Cleah Schlueter and Sharon Burks who were always willing to listen, comfort and help. Jim Stichnoth was crucial to me and helpful in countless ways, but I wish in particular to thank him for setting up the home computer environment on which I worked in the last few months of my thesis. But for the slight tendency to try to convince me to use nifty new software packages (a few of which were actually helpful), he was the best and kindest support person I could imagine.

Finally, I thank my father, Yona Talmor, who rigorously read the whole text and fished out many mistakes.

Contents

Chapter 1

Introduction

Numerical methods that solve a partial differential equation over a domain Ω, such as the finite element method and the control volume method, discretize the domain into a mesh of simple elements as a step towards a numerical solution.

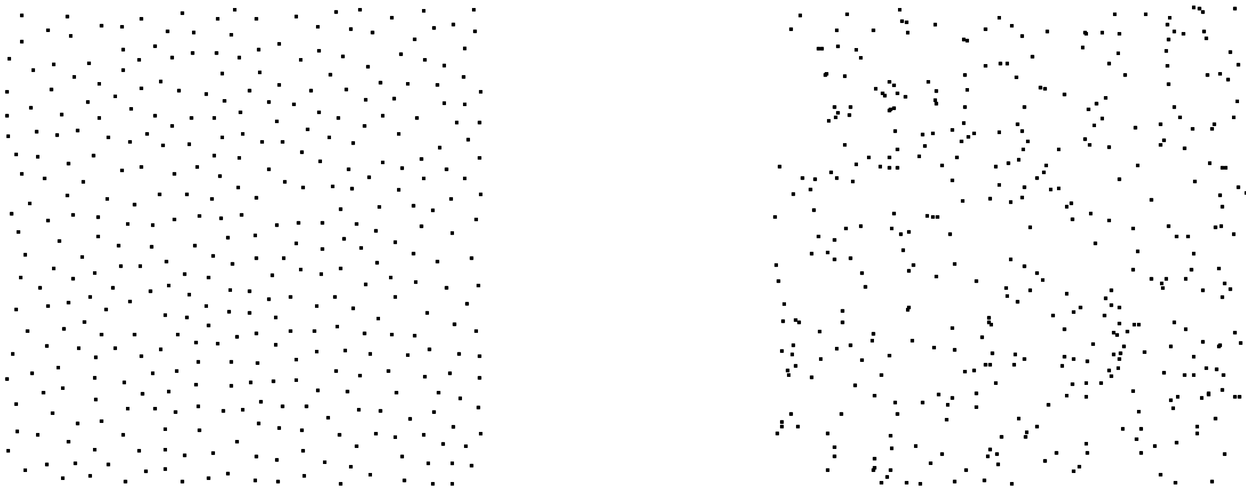
The accuracy and convergence properties of the numerical method depend on the size and shape of the mesh elements. A recent development are mesh generation algorithms that generate meshes with theoretical guarantees on the elements' shape and size. Examples for such algorithms can be found in the work of Bern, Eppstein and Gilbert [9], Chew [21], Ruppert [64], Mitchell and Vavasis [52], Miller, Talmor, Teng, Walkington and Wang [50] and Shewchuk [67]. These algorithms take as input a piecewise linear description of the domain, and partition it into triangles in two dimensions, or tetrahedra in three dimensions. The elements' shape quality is guaranteed to surpass a fixed constant lower bound; the measure used for the element quality is either its aspect ratio, in [9, 21, 64, 52] or its radius-edge ratio, in [50, 67]. Furthermore, the number of elements is guaranteed not to exceed by more than a constant factor the number of elements in any other high quality partitioning of the same domain. Following Bern, Eppstein and Gilbert, algorithms that possess these theoretical guarantees are called here *provably good mesh generation algorithms*.

This work takes one step back from provably good mesh generation. Instead of concentrating on the mesh elements shape, it focuses on the geometrical properties of the point sets used by the provably good mesh generation algorithms mentioned above. This work proceeds from discussing ways to characterize point sets which lead to provably good meshes, to methods for generating such point sets, and finally, to ways of manipulating them. This approach leads both to better understanding of the geometrical and combinatorial properties of provably good meshes, and to the development of simple and efficient algorithms for the different stages of the numerical solution of partial differential equations.

In this introductory chapter both the concept of well-spaced points and the numerical methods for which they are targeted are introduced. Some attempts are made to connect them; however, it is really the rest of this work which stitches the two together.

Section 1.1 starts with an informal discussion of well-spaced points — the concept will be formalized in later chapters. The connections between well-spaced points and the related concepts of disjoint disk models and discrepancy are then discussed. These concepts influenced and motivated the development of well-spaced points, therefore the discussion pays tribute to the chronological order in which this work evolved but is not directly related to the rest of the thesis.

Section 1.2 describes some of the numerical methods to which well-spaced points can be applied. Necessarily, the description is much simplified. The emphasis here is on giving basic background to



(a) quasi-uniform well-spaced points (b) uniformly distributed points

Figure 1.1: Contrasting well-spaced points and uniformly distributed points

the numerical methods in question, and on trying to outline some of the growing knowledge of the connection between the geometry of the mesh and the numerical properties of the solution methods.

Finally, Section 1.3 outlines the results presented in the rest of the document.

1.1 Well-spaced points

The simplest instance of well-spaced points are uniform well-spaced points; Figure 1.1(a) plots one such set. Notice that no two points are too close to each other and that there are no large empty areas where no points appear. These are the two main characteristics of well-spaced points.

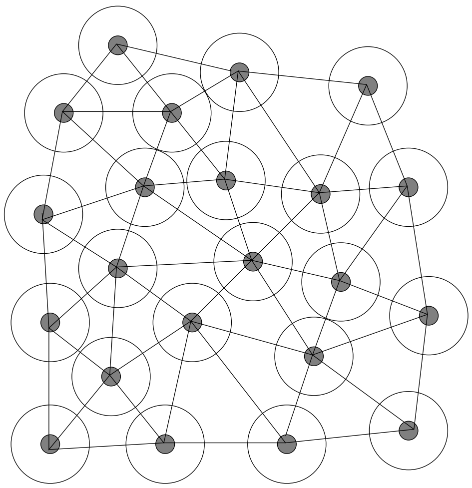
In contrast, the point set of Figure 1.1(b) contains both clusters and empty gaps, though the clustering effect is much more pronounced. This point set was constructed by choosing the two coordinates of each point uniformly at random from the unit interval.

Well-spaced points can be defined with the help of auxiliary balls used in two ways:

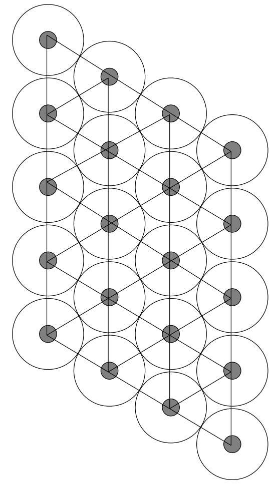
- **controlling clustering:** To prevent clustering, an auxiliary ball, called a *point ball*, is associated with each point and the set of such balls is required to be a disjoint set. A point ball is of radius r and is centered at the point. In the case of uniform well-spaced points, r is fixed and can be picked according to the desired underlying spacing between points, or simply assigned half the smallest distance between any two points.

- **controlling gaps:** auxiliary balls are associated with gaps too. To distinguish them from the point balls, they are referred to as *trial balls*. A trial ball can be centered at any point of the domain that is not inside a point ball, i.e., the center of the trial ball is at least at distance r from any point of the point set. The radius of the ball is the largest radius R such that the trial ball is disjoint from all point balls. Let R_M be the largest radius of all the trial balls. A point set is called well-spaced if R_M is not much larger then r, i.e. if for some constant C, $R_M < Cr$.

  The smaller C is, the better the quality of the well-spaced points. At the very extreme, when $C \leq 1$, the set is a *maximal well-spaced* point set, since no point can be added to it without



(a) maximal point set



(b) maximal point set, more tightly packed

Figure 1.2: Two maximal point sets: no ball can be added to either point set. Note however, that the point set at (b) is more tightly packed, and also that its minimal angle is larger.

violating the first condition.

A maximal uniform well-spaced point set has one property crucial to mesh generation: it can be triangulated with extremely well-shaped elements. In particular, the smallest angle of its Delaunay triangulation is greater than or equal to 30°. Look at Figure 1.2, showing examples of maximal well-spaced point sets. Their triangulations contain only angles larger than 30°; however, it is clear that in areas where the balls are more tightly packed, perhaps kissing, the angles are even better and approximate 60°.

Uniform well-spaced points are the simplest instance of well-spaced points; the concept can be generalized to handle non-uniform points, as in Figure 1.3. Notice that locally, the points seem uniformly well-spaced; globally, the scale of the distances between points change.

The general concept of well-spaced points is inseparable from the notion of a background spacing function, or a local feature size function, as it is sometimes called. This function describes the local scale or unit length as it changes across the domain. For uniform well-spaced points, the spacing function is a constant; referring to the point balls representation, the spacing function is r. In general, any slowly changing function can serve as a background function for well-spaced points. This imitates the slow change in element size across the domain of a good aspect ratio mesh. The concept of slowly changing functions is formalized in a later section (Definition 2.2.1) using Lipschitz functions. For a spacing function with a continuous, or piecewise continuous derivative, this definition is equivalent to requiring that the magnitude of the derivative, where it exists, be smaller than or equal to one. This restriction still allows for interesting point sets: the example of Figure 1.3 shows points converging at the center.

Uniform well-spaced points allow for good aspect ratio triangulations. This property carries over to general well-spaced point sets. Chapter 2 shows the following result for the Delaunay triangulation of any well-spaced point set: the bound on the minimal angle of the triangulation is a constant depending on the quality of the point set, that is, the size of its empty gaps. In fact, as is shown later in the thesis, the family of well-spaced point sets is the most general family allowing for bounded aspect ratio meshes.

So far in this discussion, the boundaries of the domain were not mentioned, and therefore infinite domains were implicitly assumed. However, the definition of well-spaced points can also be extended to accommodate finite domains with piecewise linear boundaries. In that case, the boundaries are treated as a one-dimensional version of a domain, and the subset of the points on the boundary segments is required to be well-spaced with respect to the lower dimensional setting of the boundaries. Furthermore, point balls are restricted to be disjoint from boundary segments they do not reside on. The issues created by the boundaries are given careful consideration in later chapters.

The development of the concept of well-spaced points, in particular uniform well-spaced points, owes a debt of gratitude to work in graphics on point set samples for rendering images. It should be emphasized that the two applications are very different: point sets for mesh generation and point samples for the integral rendering equation have different requirements. Nonetheless, the two problems are related, and since point set samples for the rendering equation were one of the motivating ideas for well-spaced points, the following two subsections discuss this relationship.

1.1.1 Disjoint disk models

Uniform well-spaced points are defined using disjoint disks. This connects the concept of well-spaced points to *disjoint disk models*. Ripley [61] discussed several disjoint disk models. For example, bearing some pre-fixed radius R in mind:

- ***Poisson disk distribution:*** This distribution is obtained by sampling N points from a Poisson process, checking if all the associated disks are disjoint, and if not, discarding the points and repeating.

- ***Poisson dart throwing:*** This family of models sample a Poisson process to obtain a set of candidates for the centers of the disks. Given this set of candidate disks, disks can be deleted to derive a disjoint disk set, in one of several ways:

  - delete all points that have a point closer than $2R$ to them, even if that point is deleted as well.

  - delete all points that have a point with an earlier birth date which is closer than $2R$ to them.

The Poisson disk distribution is hard to generate, hence the prevalence of the above, easier to generate, dart throwing models.

Disjoint disk models are useful, for example, in modeling botanical problems such as the growth of seedlings. A more recent application for disjoint disk models is in graphics, using the ray-tracing technique to render images. At the heart of ray-tracing is a point sampling method, which solves the integral rendering equation by sampling the radiance field. The number of samples and the quality of the sampling point set influence the quality of the rendered image. Yellott [74] found that the Poisson disk distribution occurs in the retinal cells of monkeys. The distribution of receptors looks very much like the distribution of points in Figure 1.1(a). If one believes that nature optimally designs, this is a strong motivation for using such samples for the rendering problem.

Mitchell [51] and Yellott [74], among others, discuss how to choose point sets to avoid problems such as aliasing in the rendered image. Their approach is from the signal processing point of view, and they identify good samples using their spectrum. In particular, the spectrum should be noisy, and have a deficiency of low-frequency energy. This is called the *blue-noise criteria*. Yellott recommends

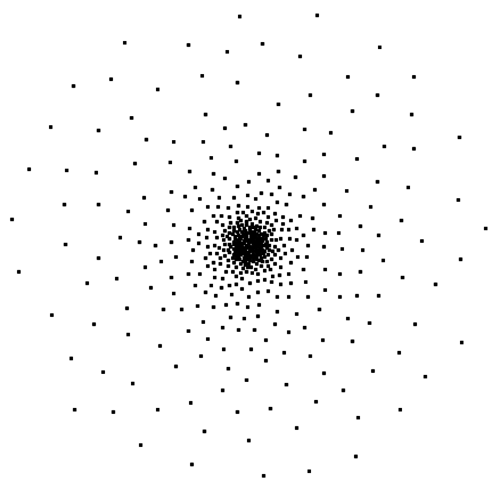


Figure 1.3: Non-uniform well-spaced points

the Poisson disk distribution as a distribution possessing the blue-noise criteria. Mitchell suggests other, simpler to generate, disjoint disk models, that also satisfy the blue-noise criteria.

Relationship between disjoint disk models and well-spaced points: The point set of disjoint disk samples can be a well-spaced point set if they have no large gaps. However, restricted gaps are not an explicit requirement of disjoint disk models. Rather, if the size of a sample is large enough, it can be argued that the probability of large gaps occurring is extremely low.

On the other hand, the family of uniform well-spaced point sets includes sets that are inappropriate for the applications in which disjoint disk models are used. For example, the nodes of a uniform grid constitute a very good well-spaced point set for mesh generation purposes, but it is a poor point set for rendering images in that it is more prone to aliasing problems. Mitchell [51] gives an example of an image where uniform grid samples cause the appearance of large Moiré patterns, but Poisson disk samples produce a much higher quality image.

Another important difference between the two concepts is the ways in which they have been generalized. Ripley [61] discussed assigning a different radius to each point sample. The radius is to be taken from some prescribed distribution. Two rules for deleting points are then suggested: deleting a point if its disk intersects some other disk, or, deleting a point if it falls within some other disk. The generalization we use, restricting the disk sizes to behave as a slowly changing function, is very different in spirit.

1.1.2 Discrepancy

Shirley [68] was the first to introduce *discrepancy* as a quality measure of samples for the rendering equation. The concept of discrepancy is from the field of numerical integration, where the problem of identifying good sample point sets has been researched in depth. This subsection discusses point sets of low discrepancy and their connection to well-spaced points. Niederreiter [56] provides an excellent introduction to discrepancy, and this short discussion of discrepancy is mostly based on his presentation. Another more recent source is the book by Drmota and Tichy [27].

Let $I_d = [0,1]^d$ be the unit cube in d dimensions, and let S be a family of subsets of I_d. The discrepancy of a point set $P \subset I_d$ with respect to S is

$$D(S;P) = \sup_{s \in S} \left| \frac{|s \cap P|}{|P|} - A(s) \right|,$$

where $A(s)$ is the area or measure of s, and $|Q|$ is the number of points in Q.

Some commonly used families of sets are:

- $\mathcal{J}^*$: the family of subintervals of the form $\Pi_{i=1}^{d}[0, u_i)$.
- $\mathcal{J}$: the family of subintervals of the form $\Pi_{i=1}^{d}[u_i, v_i)$.
- $\mathcal{C}$: the family of convex subsets of I_d. The resulting discrepancy is called *isotropic discrepancy*.

The concept of discrepancy arises in numerical integration. In order to numerically approximate an integral of a function over a one-dimensional interval, trapezoidal-like rules similar to the following are used:

$$\int_0^1 f(u)du \approx \sum_{n=0}^{N} w_n f\left(\frac{n}{N}\right).$$

The error of this approximation is $O(\frac{1}{N^2})$ under certain smoothness conditions on f. The problem is that these rules do not generalize very well to higher dimensions, and in d dimensions have errors of the order $O(\frac{1}{N^{2/d}})$. This problem (known as "the curse of dimensionality") has lead to the development of Monte Carlo integration methods. The Monte Carlo estimate for the integral uses N independent random samples $x_1, \ldots, x_N$ from the interval

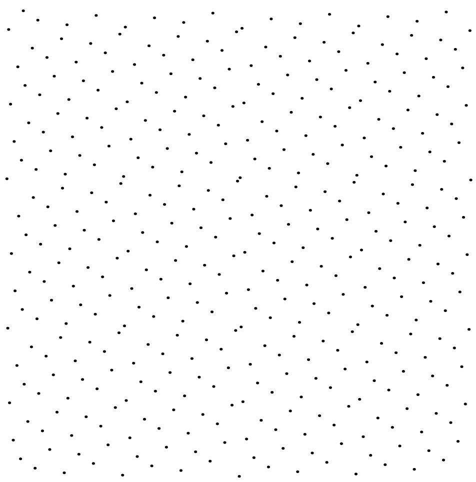
$$\int_0^1 f(u)du \approx \frac{1}{N} \sum_{n=0}^{N} f(x_n).$$

The error of this method is estimated probabilisticly: the expected absolute value of the error is $O(N^{-1/2})$, with a constant coefficient dependent on the variance of the function f. The advantage is that this error estimate holds in higher dimensions as well, so the Monte Carlo integration method does not deteriorate with increasing dimensions. However, the fact that the estimate is probabilistic means that there exist point samples that obtain even smaller error, since the above error estimate was the average of the error over all point samples. Can such point sets be constructed deterministically? The study of this question leads to point sets of low discrepancy.

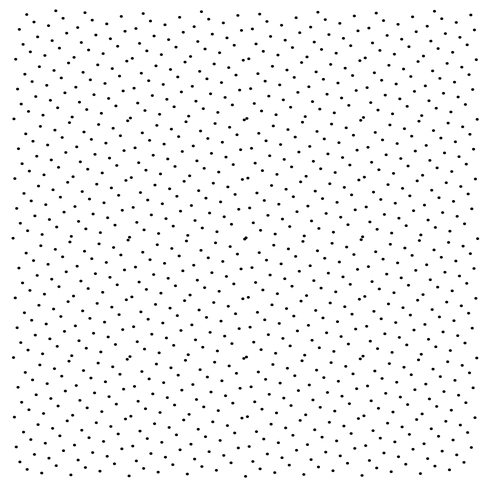
Point sets of low discrepancy can be constructed deterministically, and for a function f of bounded variation $V(f)$, the integration error over I_d using the estimate derived from a point set P is bounded by $V(f)D(\mathcal{J}^*, P)$. Furthermore, it is possible to construct point sets P of discrepancy $D(\mathcal{J}^*, P) = O((\log N)^{d-1}/N)$. This is a great improvement over both the Monte Carlo method and trapezoidal-like integration rules.

In one or two dimensions, any point set with N points must obey $D(\mathcal{J}^*, P) = \Omega((\log N)^{d-1}/N)$. The Hammersley sequence defined below and plotted in Figure 1.4 achieves this lower bound. Note that the Hammersley sequence is a finite sequence, and that the Hammersley sequence of length N is not a prefix of the Hammersley sequence of length $N+1$.

For infinite sequences, it is conjectured that the optimal discrepancy is of order $\Theta((\log N)^d/N)$, however this conjecture is open for $d \geq 2$. The *Halton sequence* defined below and shown in Figure 1.5, achieves this bound.

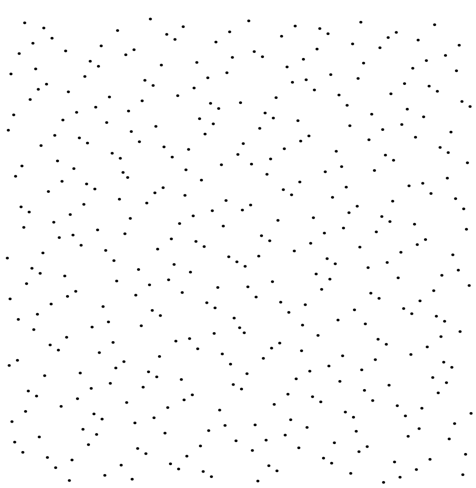


(a) 400 points

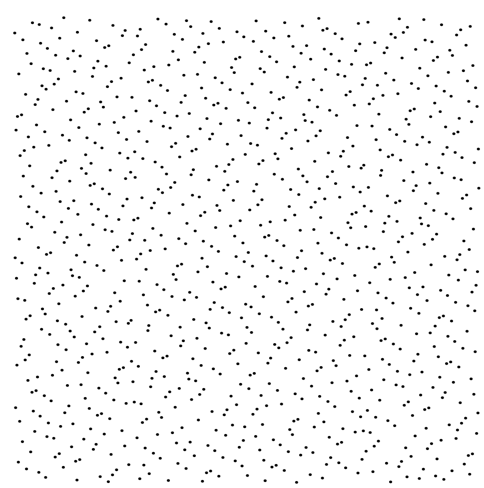


(b) 1000 points

Figure 1.4: A two dimensional Hammersley sequence with $b_1 = 2$.



(a) first 400 points



(b) first 1000 points

Figure 1.5: A two dimensional Halton sequence with $b_1 = 2$ and $b_2 = 3$.

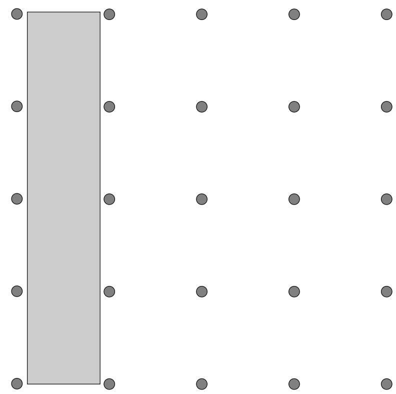


Figure 1.6: Points arranged in a grid have high discrepancy, since a rectangle of area $\frac{1}{\sqrt{N}}$ can be empty.

Definition 1.1.1 (Hammersley sequence) *Given a set of* $d-1$ *relatively prime integers* $b_1, \cdots, b_d$*, and* N*, the length of the sequence, the* n*-th element in the Hammersley sequence is*

$$H(n) = \left(\phi(n, b_1), \cdots, \phi(n, b_{d-1}), \frac{n}{N}\right).$$

If the representation of n *in base* b *is* $n = \sum_{j=0}^{\infty} a_j b^j$ *then* $\phi(n, b) = \sum_{j=0}^{\infty} a_j b^{-j-1}$*. (*$\phi$ *simply inverts the representation of* n *in base* b *around the period, transforming the integer into a number between zero and one).*

Note that this sequence is of length N*, and can not be extended.*

Definition 1.1.2 (Halton sequence) *Given a set of* d *relatively prime integers* $b_1, \cdots, b_d$*, the* n*-th element in the Halton sequence is*

$$H(n) = (\phi(n, b_1), \cdots, \phi(n, b_d)).$$

Relationship between low-discrepancy sequences and well-spaced points: The most important difference between the two is that in a low-discrepancy sequence, points can get arbitrarily close. This is clear from looking at the Halton sequence of Figure 1.5 and the Hammersley sequence of Figure 1.4, but also can be concluded from the definition of discrepancy: given N points, assume a pair of points are very close. This adds at most $1/N$ to the discrepancy, and obviously the sequence can still be a low-discrepancy sequence.

On the other hand, a set can be well-spaced and still have far from optimal discrepancy; a simple example is N points in a grid pattern, see Figure 1.6. A rectangle of width $1/\sqrt{N}$ and height one can be placed so that it contains no points, resulting in discrepancy larger than $1/\sqrt{N}$ and far from optimal. Note that a similar example can be constructed when the test set is a circle rather than a rectangle, by taking a circle of radius $1/2$ and constructing a well-spaced point set that has no points in a ring of width $1/\sqrt{N}$ placed inside the circle and around its circumference. The discrepancy is again of order $1/\sqrt{N}$, the area of the empty ring. This example is important, since it seems natural to measure discrepancy with respect to circles, thus relating it to the gaps of well-spaced points.

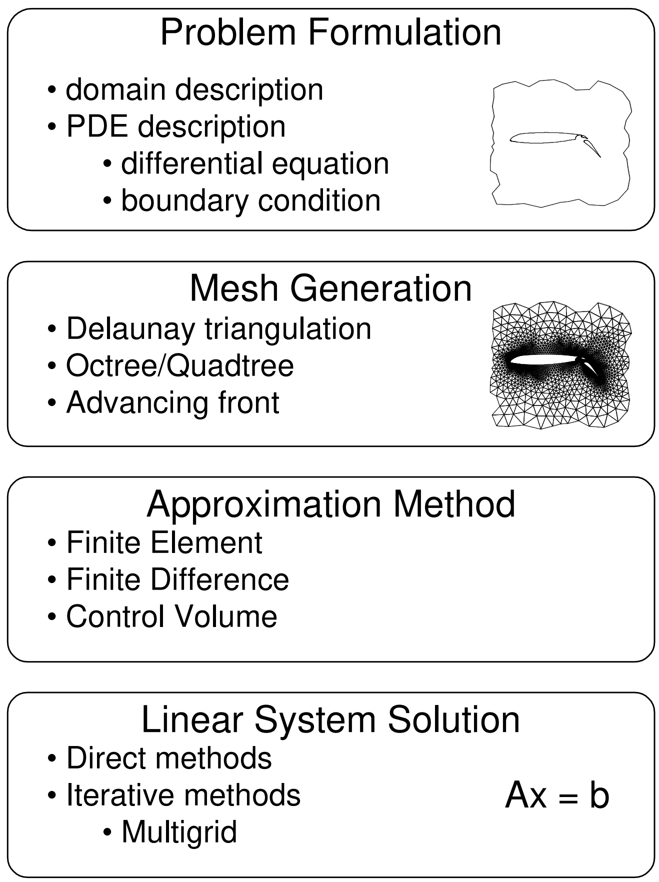


Figure 1.7: A template for a numerical method for the solution of a PDE

1.2 Numerical methods

Given a physical domain Ω and a physical problem to solve over the domain, such as heat flow, a numerical method may take the following steps to solve the problem:

1. **Problem formulation:** the problem is phrased as a partial differential equation and the continuous domain is approximated using a simpler, discrete description. In this work domain descriptions are assumed to be of the simplest form — piecewise linear. However, in general the boundary may contain curved elements.

2. **Mesh generation:** the interior of the domain is decomposed into a mesh M of simple and well-shaped elements. Here, the focus is on unstructured meshes of triangular or tetrahedral elements. Some methods discretize the domain using only points [42]. The finite difference method can be developed using points and their neighborhoods only, the neighborhood of the point being a finite set of points close to it. However, the standard formulation of the method uses a grid of points, and the neighborhood of each point is composed of its grid neighbors. In that case the neighborhood is referred to as a "stencil."

3. **Approximation:** a system of linear or non-linear equations is formed over M to approximate the governing PDEs (e.g., the stiffness matrix and the right hand vector are assembled).

4. **Linear system solution:** the system of equations is solved, and the error of the solution estimated. An exception is the multigrid method, which requires the formulation of a sequence of meshes, and a sequence of linear systems, as part of the solution process.

The first step is that of modeling the physical problem using a partial differential equation. For example, the general form of a linear partial equation in two dimensions is

$$A\frac{\partial^2 u}{\partial x^2} + B\frac{\partial^2 u}{\partial x \partial y} + C\frac{\partial^2 u}{\partial y^2} + D\frac{\partial u}{\partial x} + E\frac{\partial u}{\partial y} + Fu + G = 0$$

The coefficients can be scalars or functions over the domain. Partial differential equations (PDEs) can be divided into the following three categories, depending on A, B and C:

- elliptic: $B^2 - 4AC < 0$

- parabolic: $B^2 - 4AC = 0$

- hyperbolic: $B^2 - 4AC > 0$

This classification is important because different categories of equations require different boundary conditions and lend themselves to numerical solving by different methods. The literature on PDEs and the numerical solution of PDEs is rich and varied; see for example Fletcher [29], Becker et al. [7], Johnson [38], and Carey [14].

Figure 1.7 gives a template for the numerical solution of partial differential equations. The template is appropriate for methods such as as the finite element, control volume method, and to some extent the finite difference method.

1.2.1 The finite element method

The presentation here closely follows Strang's [69] introduction to the finite linear element method.

The finite element method approximates a solution function using a combination of a finite number of basic functions — polynomials. This is in contrast to the finite difference method, which obtains an approximation to the function values at a finite set of points.

One way to develop the finite element method is based on the weak form of a partial differential equation (PDE). Consider the following PDE, Poisson's equation

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = f(x, y),$$

and the boundary condition:

$$u = 0 \quad \text{on the boundary.}$$

Poisson's equation is the simplest elliptic equation. To derive its weak form, multiply the equation by a test function $v(x, y)$ and integrate over the domain:

$$\int\int_\Omega \left[-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} \right] v(x, y) dx\, dy = \int_\Omega f(x, y) v(x, y) dx\, dy$$

Integrating the left side by parts, and assuming that v vanishes on the boundary (as u does) the more symmetric form is obtained:

$$\int\int_\Omega \left[\frac{\partial u}{\partial x}\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial v}{\partial y} \right] dx\, dy = \int_\Omega f(x, y) v(x, y) dx\, dy$$

This weak form holds for every test function v. To obtain a numerical method, a finite set of trial functions is chosen: $T_1(x, y), \ldots, T_n(x, y)$. These functions are used in two ways:

1. The function U which is the approximation of u is assumed to be $U = \sum_i U_i T_i$, where the U_i's is a set of unknown constants.

2. The test function v is assumed to be one of $T_1, \ldots, T_n$

For each i, use the weak form and the assumption that the test function v is T_i to obtain the following set of equations:

$$\int\int_\Omega \left[\frac{\partial U}{\partial x}\frac{\partial T_i}{\partial x} + \frac{\partial U}{\partial y}\frac{\partial T_i}{\partial y} \right] dx\, dy = \int_\Omega f(x, y) T_i dx\, dy$$

Now, use the assumption that U, the approximation of u, can be written as a sum $U = \sum_i U_i T_i$, and substitute this sum in the equation above to obtain an $n \times n$ linear equation $K\hat{U} = \hat{f}$ where $\hat{U}$ is the vector of the unknowns U_i, $\hat{f}$ is the vector with entries of the form

$$\hat{f}_i = \int_\Omega f(x, y) T_i dx\, dy,$$

and the entries of K are of the form

$$K_{ij} = \int\int_\Omega \left(\frac{\partial T_i}{\partial x}\frac{\partial T_j}{\partial x} + \frac{\partial T_i}{\partial y}\frac{\partial T_j}{\partial y} \right).$$

The simplest form of an element in two dimensions is obtained when the domain is divided into triangles. The trial functions are piecewise polynomials, and are strictly polynomial on each triangle. The simplest form of a trial function is a linear polynomial on each triangle. In that case, the trial function T_i is chosen to be 1 at node i, and 0 at all other nodes. This means that T_i will be non-vanishing only on a triangle incident to node i. Consider the triangle $\Delta(i,j,k)$, and let θ_i stand for the triangle angle at node i. The contribution of the triangle $\Delta(i,j,k)$ to the matrix K can be written in terms of the angles θ_i, θ_j and θ_k:

$$K(\Delta(i,j,k)) = 0.5 \begin{bmatrix} \cot\theta_k + \cot\theta_j & -\cot\theta_k & -\cot\theta_j \\ -\cot\theta_k & \cot\theta_i + \cot\theta_k & -\cot\theta_i \\ -\cot\theta_j & -\cot\theta_i & \cot\theta_i + \cot\theta_j \end{bmatrix}$$

The matrix $K(\Delta(i,j,k))$ is referred to as the local stiffness matrix of the triangle. Its entries were computed by using the fact that $\frac{\partial T_i}{\partial x}$ and $\frac{\partial T_i}{\partial y}$ are constant over the element for linear T_i's. The matrix K, also called the global stiffness matrix, is assembled by adding the contributions from all the local matrices at the appropriate entries.

In three dimensions, the 4×4 local matrix associated with a tetrahedron is similar. Instead of angles at a node, dihedral angles at an edge are used. Let h_{kl} be the length of edge (k,l), and let θ_{kl} be the dihedral angle at the edge, then the (i,j)-th entry of the local matrix equals

$$K(\Delta(i,j,k,l))_{(i,j)} = -h_{kl}\cot\theta_{kl}$$

and as in two dimensions, the diagonal entries complete the row so the sum of all the entries in the row is zero. Note, that in two dimensions the lengths of the edges do not affect the stiffness matrix, whereas in three dimensions each entry carries a multiplier whose magnitude depends on the edge length. In one dimension, the stiffness matrix depends on one over the edge length.

1.2.2 Connecting geometry and numerical errors

Knowing what defines a good mesh is the first step in designing a provably good mesh generation algorithm. This statement seems almost too obvious to elaborate upon. Sadly, however, for many numerical methods the geometric conditions have not been explicitly worked out. The situation for the finite elements method is somewhat better in that regard, but still far from perfect — as this section discusses.

No large angles: In their famous paper, Babuška and Aziz [1] show that the finite element method converges when the triangle sizes decrease, as long as the largest angle is bounded away from 180°. Furthermore, they give an example where using a triangulation with small angles converges but using a triangulation with large angles (as well as small angles) fails to converge. Thus, to guarantee convergence of the finite element method over a triangular mesh, the mesh should contain no large angles.

No small angles: If mathematical correctness of the method was all we cared about, it would suffice to require the mesh to contain no large angles. However, element shape can also affect the work necessary to obtain a solution to the linear system. In particular, both large angles and small angles can result in ill-conditioned, and therefore harder-to-solve, matrices.

The effect can be seen clearly in the case of Poisson's equation. For other partial differential equations, this may not be the case. This state of things is what makes obtaining general geometric requirements so difficult.

Looking at the explicit formulation of the stiffness matrix entries derived above for Poisson's equation, it is clear that when $\theta \to 0$, then $\cot\theta \to \infty$. Large entries in the matrix can lead to the matrix being ill-conditioned, i.e. of large condition number $\kappa(K)$. Large condition numbers affect the solution time or solution accuracy of the linear system.

- **direct solution:** when solving the linear system directly, the number of accurate digits in the solution depends on the condition number of the system. For example, the following heuristic appears in [32]: if the unit roundoff is $\sim 10^{-q}$, then Gaussian elimination produces a solution that has about $q - \log_{10}\kappa(K)$ correct decimal digits.

- **iterative solution:** When solving the linear system iteratively, the number of iterations necessary to converge to the solution of the system depends on the magnitude of its condition number. For example, the conjugate gradient method requires $O(\sqrt{\kappa(K)})$ steps to converge [38].

The condition number of the resulting matrix should therefore be as small as possible for the system to be efficiently and effectively solved. Notice however, that preconditioning techniques can improve the condition number of the matrix and thus countermand the effect of small angles in certain cases.

The condition number of the stiffness matrix associated with an elliptic problem of order 2, such as Poisson's equation above, is $O(h^{-2n})$, under the assumption that the triangulation is a quasi-uniform, good aspect ratio triangulation with typical edge length h and n nodes. For a proof, see Johnson [38], page 141. However, this condition number degrades if smaller angles are allowed. What is the effect of one small angle on the condition number of the matrix?

To estimate that condition number, first consider the one dimensional Poisson equation: $-u''(x) = f(x)$. Its associated stiffness matrix K_n is of the form:

$$K_n = \frac{1}{h}\begin{bmatrix} 2 & -1 & \cdot & \cdot & \cdot \\ -1 & 2 & -1 & \cdot & \cdot \\ \cdot & -1 & 2 & -1 & \cdot \\ \cdot & \cdot & -1 & 2 & -1 \\ \cdot & \cdot & \cdot & -1 & 2 \end{bmatrix}.$$

This system is the Laplacian of a line graph with n nodes, and with unit weights associated with the edges. A small angle in a two dimensional mesh is analogous to placing a higher weight on an edge. Picking one edge — the edge between nodes i and $i+1$ — let $v = \cot\theta$ be the new weight added to the edge for some small angle θ. The corresponding matrix K'_n has four modified entries:

- $K'_n(i,i) = K_n(i,i) + v$, $K'_n(i+1,i+1) = K_n(i+1,i+1) + v$, and

- $K'_n(i+1,i) = K_n(i+1,i) - v$, $K'_n(i,i+1) = K_n(i,i+1) - v$.

Note that the weight change v is subtracted from the off diagonal entries and added to appropriate diagonal entries. For a symmetric matrix A, the condition number is equal to

$$\kappa(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)},$$

where $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ are the smallest and largest eigenvalues of the matrix respectively. Clearly, for the matrix K',

$$\lambda_{\max}(K') = \max_{|x|=1} x^T K'_n x \geq v.$$

To estimate $\lambda_{\min}(K')$, use the fact that the original, unmodified matrix K_n is known to have a smallest eigenvalue which is of order $O(\frac{1}{n^2})$. Also, note that

$$x^T K_n' x = \left(\sum_{(k,l)\in \text{ edges}} (x_k - x_l)^2 \right) + v(x_i - x_{i+1})^2 + x_1^2 + x_n^2$$

Let x be some n-dimensional vector such that $x_i = x_{i+1}$. Let y be an $(n-1)$-dimensional vector formed from x by removing the i-th entry. Clearly, by the above expansion

$$x^T K_n' x = y^T K_{n-1} y.$$

The fact that for any matrix A

$$\lambda_{\min} = \min_{|x|\neq 0} \frac{x^T A x}{x^T x},$$

implies that

$$\lambda_{\min}(K_n') \leq \lambda_{\min}(K_{n-1}) = O(\frac{1}{n^2})$$

and that

$$\kappa(K_n') = \Omega(\frac{1}{\tan\theta} n^2).$$

A similar argument works for a two dimensional mesh. Therefore, the appearance of even one or very few small angles in the mesh can adversely affect the condition number in a multiplicative manner.

Other requirements: Some requirements do not explicitly concern the element shape. For example, when solving Poisson's equation it is beneficial to require the underlying triangulation to be Delaunay since then the stiffness matrix is an M-matrix. A matrix with non-positive off diagonal entries is an M-matrix if its inverse is nonnegative. Such matrices possess many nice properties which make them numerically preferable.

1.2.3 The control volume method

The control volume method uses a partition of space into small "control volumes" around each node of the mesh in order to derive the linear system whose solution approximates the function at the nodes. One way to construct these volumes is by using a Delaunay triangulation as the underlying mesh, and its dual, the Voronoi diagram, for the control volumes. The Voronoi cell of a node is composed of the set of domain points closer to it than to any other mesh node.

The presentation of the control volume method that appears below is due to Walkington, and appeared first in Miller, Talmor, Teng and Walkington [49]. Here the finite volume method for a general Delaunay diagram is described, and geometric conditions on the mesh, that guarantee convergence of the method, are outlined. There are several ways in which this exposition differs from the typical discussion of the control volume method:

- The control volume method can be defined on any triangulation, and not only the Delaunay triangulation; in that case the control volumes are not defined by the dual, but can be constructed in the following way: place auxiliary points at the center of all triangles, and at all the edge midpoints. The control volume of a particular mesh node is constructed by walking in a clockwise over all the triangles incident to the mesh node, and connecting the triangle center points to the edge midpoints in the order they are encountered.

- Often a Delaunay triangulation is used with the additional requirement that all the triangles be non-obtuse. In that case, the Voronoi edges cross their dual Delaunay edges. However, this restriction is waived here.

- The geometric condition important here is the ratio of the circumscribing sphere or circle of a Delaunay triangle to the smallest edge in the triangle, called the *radius-edge ratio*. In two dimensions, bounding this ratio is equivalent to bounding the smallest angle away from zero; in three dimensions, however, this is no longer true, and a bounded radius-edge ratio does not imply that all angles are bounded away from zero. Introducing the radius-edge ratio is one of the contributions of this thesis.

Now, for 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. Consider Delaunay triangulations of Ω satisfying the property that Voronoi regions corresponding to interior vertices are contained within Ω. The control volume technique uses both the Delaunay and the Voronoi diagrams. Letting V_i be a Voronoi region corresponding to an interior vertex p_i, integrate the partial differential equation 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. Let the length of the Delaunay edge joining vertex p_i to p_j be denoted by h_{ij}, and let $\mathcal{N}_i$ 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, is denoted by A_{ij}; see Figure 1.8. Letting u_i be an approximation of $u(p_i)$ (u being the exact solution), the above equation is approximated by:

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

In the above, f_i is the average value of f over the Voronoi region V_i. This equation holds for each interior vertex p_i, and on the boundary set $u_i = g(p_i)$. Construct a matrix for the linear system with variables u_i, such that row i of the matrix contains the above linear constraints. The solution of this matrix is the control volume approximation of our equation. Clearly, the matrix corresponding to this system of linear equation is an M–matrix, so the discrete maximum principle still holds. MacNeal [43] shows that, in two dimensions, this matrix is identical to that given by the finite element method constructed using piecewise linear functions on the triangulation; however, this is not the case in three dimensions.

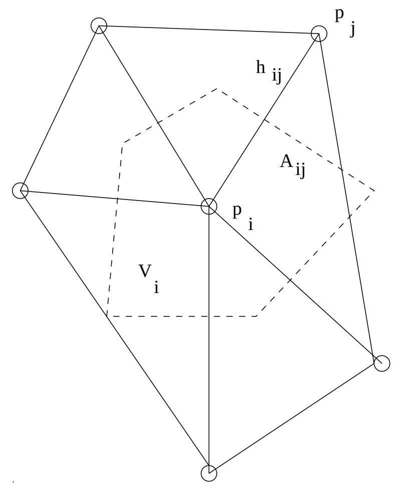


Figure 1.8: The Voronoi diagram of p_i, the Delaunay edge h_{ij}, and the Voronoi face A_{ij}.

Nicolaides [55] established error estimates for the control volume method. For quantities defined on the edges of the Delaunay mesh Nicolaides defines the inner product $(.,.)_W$

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

and corresponding norm $\|.\|_W$ by

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

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

Theorem 1.2.1 (Nicolaides) *Let $\{u_i\}$ be the discrete solution given by the control volume method, then 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 proceeds to estimate the error $\|U^{(1)} - U^{(2)}\|_W$ under the assumption that the meshes had bounded aspect ratio; however, Walkington [49] shows that this hypothesis can be relaxed to the situation where the radius-edge ratio is bounded.

Theorem 1.2.2 (Walkington) *Let $\rho = \max_{(i,j)} r_{ij}/h_{ij}$ be the worst radius-edge ratio; then*

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

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

As usual, the estimate above can be localized in the sense that the right hand side is actually sums of products of h_{ij} and the L^2 norm of the second derivatives in small regions containing this edge.

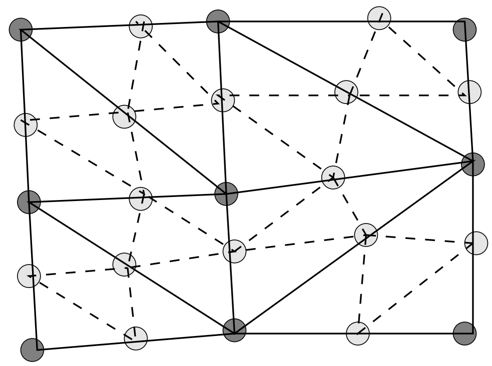


Figure 1.9: Nested coarse and fine triangulations. The new edges of the finer triangulation are dashed; its new nodes are lightly colored

1.2.4 The multigrid method

The multigrid method is an iterative method that uses a sequence of respectively coarser meshes and their associated linear systems to quickly converge at a solution for the linear system formed over the initial, fine mesh.

Brandt [12] was the first to suggest the multigrid method. He has shown that under ideal conditions, for example when the mesh is a uniform grid, multigrid is optimal in the sense that it converges in $O(n)$ steps, where n is the number of nodes in the grid.

The presentation here follows Johnson [38]. To understand the basic step of the multigrid method, consider two triangulations T_0 and T_1 as in Figure 1.9. Assume T_0 was obtained by subdividing each triangle of T_1 into four similar triangles. The two triangulations thus nest, and the number of nodes increases geometrically. Let n stand for the number of nodes of T_0; then the number of nodes of T_1 is $O(n/4)$. To solve the system A_0 associated with T_0, first assume the system A_1 can be solved efficiently at cost $O(n/4)$ linear in the number of its nodes. A step of the multigrid takes an approximation u_0^k of the solution to A_0 to an improved approximation u_0^{k+1} of A_0 using the following two basic substeps:

- **smoothing step:** m iteration of an iterative method, such as a gradient method, Jacobi, or Gauss-Seidel, resulting in an intermediate vector $u_0^{k+\frac{1}{2}}$.

- **coarse grid correction:** obtaining a correction to the intermediate solution $u_0^{k+\frac{1}{2}}$ by reducing the problem to the coarse grid. For the required correction y, it holds that $A_0 y = b - A_0 u_0^{k+\frac{1}{2}}$. Given operators to first transform the problem onto the coarse grid, an approximation to the correction y can be obtained on the coarse grid at cost $O(n/4)$ and then transformed back. u_0^{k+1} is then defined as $u_0^{k+\frac{1}{2}} + y$.

Under suitable assumptions it can be shown that one multigrid step reduces the error significantly. The intuition is that the smoothing step is very effective in reducing the high frequency components of the error, whereas the coarse grid is effective in reducing the medium and low frequency components of the error, since on the coarser grids they appear more high-frequency like. This intuition is wonderfully illustrated in Briggs' [13] short multigrid tutorial.

The next step is that of generalizing this simple multigrid method to unstructured meshes, and then to non-nested sequences of respectively coarser meshes. This is inevitable in the solution of complex problems with more intricate domain geometry and solutions. The theory behind applying multigrid to unstructured meshes is not as well understood as it is for the structured and nested case. However, it is an area of active research, with increasing theoretical progress [11, 15, 17]. In practice, multigrid methods on unstructured, non-nested mesh sequences are popular despite this current lack of full theoretical foundations [6, 57].

The effectiveness of a multi-level method that uses an unstructured coarsening sequence of meshes $M_0, \ldots, M_k$ depends on the quality of this sequence [11, 15, 17]. In particular, Chan and Zou [17] provided sufficient conditions for multi-level additive Schwarz methods to work on unstructured meshes. Informally, their conditions require, for each i in the range $1 \leq i \leq k$, that

- M_i is well-shaped, e.g., in two dimensions elements of M_i should have a bounded aspect ratio; and
- M_i approximates M_{i-1} in the numerical formulation.

This thesis addresses, among other issues, the question of generating mesh coarsening sequences that are guaranteed to have the above properties.

1.3 Thesis Outline

The rest of the thesis is divided into two parts:

- part I: **well-spaced points**
- part II: **mesh coarsening**

The first part defines and discusses well-spaced points in two and three dimensions. The second part gives a particular application for well-spaced points, that of mesh coarsening for the multigrid method. The mesh coarsening problem, algorithm, and experiments are presented in two dimensions and therefore rely on Chapter 2 where properties of two dimensional well-spaced points are discussed and proven.

Chapter 2 introduces the concept of well-spaced points, and proves that it is equivalent to the concept of a bounded aspect ratio mesh in two dimensions. In particular, the following is shown:

- Every bounded aspect ratio mesh is composed of well-spaced points. Furthermore, interior domain points are not too close to the boundary in terms of the local spacing.
- The constrained Delaunay triangulation of a well-spaced point set, where no point of the set is too close to a non-incident boundary segment in terms of the local spacing, is a bounded aspect ratio mesh conforming to the domain.

This equivalence lays the foundations for the rest of the thesis. It enables one to develop algorithms for mesh coarsening and mesh generation that focus on the underlying well-spaced point set of a mesh.

Chapter 3 generalizes to three dimensions the first part of the results of Chapter 2: that is, the node set of a good aspect ratio mesh in 3D is a well-spaced point set. Furthermore, the chapter introduces the radius-edge ratio and shows that bounded radius-edge ratio Delaunay meshes have the following properties:

- their 1D skeleton is a density graph,
- their Voronoi cells are well-shaped, and
- their nodes are well-spaced points.

Bounded radius-ratio Delaunay meshes are composed of good aspect ratio elements and sliver elements. A sliver element is a flat, thin element whose edge lengths and circumradius are all similar. Delaunay meshes with bounded radius-edge ratios are important in the context of the control volume method, and because the Delaunay triangulation of well-spaced points leads to a mesh with bounded radius-edge ratios, but does not lead to bounded aspect ratios since it allows for slivers. In some sense, the equivalent of well-spaced points in three dimensions is a bounded radius-edge ratio mesh.

Chapter 4 presents an algorithmic framework for generating a bounded aspect ratio Delaunay mesh conforming to a given piecewise linear non-manifold domain. The algorithm proceeds by packing points on the different boundary elements in increasing order of dimension and finally in the interior. It is a step towards showing that well-spaced points lead to bounded radius-edge ratio meshes in three dimensions.

The rest of the thesis concentrates on mesh coarsening in two dimensions. The problem of mesh coarsening is that of creating a sequence of respectively coarser meshes from an initial bounded aspect ratio mesh. All the meshes in the coarsening sequence should have bounded aspect ratios. A mesh in the coarsening sequence is a smaller approximation of its preceding mesh, in the sense that it contains less elements and nodes.

Chapter 5 addresses the problem of mesh coarsening theoretically, by first introducing the function-based coarsening approach, and then showing an algorithm based on this approach can be designed to be optimal in the following sense: (1) the sequence it constructs is a bounded aspect ratio sequence satisfying the restriction of mesh coarsening mentioned at the end of Section 1.2.4 and (2) is the smallest such sequence (up to a constant factor) both in terms of the sequence length and the size of the meshes in the sequence. Furthermore, the chapter shows that this optimal sequence can be constructed in time linear in the total size of the sequence meshes.

The function-based approach is contrasted with a popular topological approach we term the MIS-based method. This method creates a coarsening sequence by picking a maximal independent set (MIS) of a finer mesh to form the next coarser mesh in the sequence. Once the coarser mesh nodes are selected using the MIS, the coarse mesh is generated by triangulating the nodes, e.g. using the Delaunay triangulation. The MIS-based method is an iterative coarsening method that is subject to the problem of repeated degradation.

Chapter 6 discusses the problem of repeated degradation in *iterative coarsening* methods. An iterative coarsening method generates the mesh at level M_{i+1} using only the information embedded in the mesh M_i of the previous finer level. Iterative coarsening is particularly vulnerable to the problem of repeated degradation: coarsening algorithms usually degrade the quality of the coarser mesh somewhat; for example, the aspect ratio of the elements becomes a little worse. As the process is repeated, the quality of the coarser meshes eventually plunges below acceptable levels.

Chapter 6 also addresses the iterative application of two coarsening methods: MIS-based coarsening and function-based coarsening. It should be noted that MIS-based coarsening must always be applied iteratively, whereas function-based coarsening can be applied non-iteratively as well and thus bypass the question of repeated degradation altogether.

The MIS-based method is shown here to be subject to repeated degradation even in the case of quasi-uniform meshes. Function-based coarsening methods, even in their iterative application, are shown to be immune to repeated degradation over quasi-uniform meshes.

Chapter 7 presents a practical algorithm for mesh coarsening that is based on the theoretical function-based coarsening framework developed in Chapter 5. Several modifications of the theoretical algorithms lead to a simple and efficient algorithm that is easy to implement and generates the mesh M_i of the coarsening sequence at cost $O(n_i \log n_i)$.

Finally, Chapter 8 tests an implementation of the practical algorithm over a test suite of highly graded meshes and contrasts it with an implementation of the MIS-based approach. The practical algorithm is shown to produce high-quality coarsening sequences.

The thesis concludes with a discussion of the open problems and future work.

Part I

Well-Spaced Points

Chapter 2

Well-Spaced Points and Well-Shaped Meshes in Two Dimensions

This chapter introduces and discusses the concept of well-spaced points in two dimensions. Well-spaced points were introduced informally in the previous chapter using the following two properties: points are not too close to each other, and they also do not stray too far relative to the local spacing size. This chapter formally defines the concept of well-spaced points and shows they are equivalent to bounded aspect ratio meshes in two dimensions.

Section 2.1 gives the basic definitions of two-dimensional domain descriptions, meshes, and measures of mesh qualities. Section 2.2 describes spacing functions and mesh functions used to quantify the local spacing. Section 2.3 formally defines the notion of well-spaced points. Section 2.4 shows each bounded aspect ratio mesh is composed of well-spaced points, and Section 2.5 shows that a well-spaced point set can be used to generate a bounded aspect ratio mesh.

2.1 Domains and meshes

A **Planar Straight Line Graph (PSLG)** is a collection of line segments and points in the plane, closed under intersection. A **domain** Ω is a planar straight-line graph (PSLG), whose boundary is a collection of polygons (i.e., holes are allowed). The PSLG is a linear model of some continuous domain in the plane, possibly containing internal edges and points representing boundaries between two materials, points of special interest, or holes.

A two-dimensional **mesh** is a discretization of the geometric domain described by a PSLG into simple polygonal **elements**. The intersection of two elements must either be an edge of both elements, or a point of both. The **mesh nodes** are the vertices of its elements. A mesh is referred to here using the notation $\mathcal{M} = (\mathcal{P}, E, \mathcal{B})$ or $\mathcal{M} = (\mathcal{P}, \mathcal{T}, \mathcal{B})$. $\mathcal{P}$ is the embedded node set of the mesh $\mathcal{M}$. $\mathcal{T}$ is the element set, and E is the edge set. $\mathcal{B}$ is the boundary PSLG description. Nodes from $\mathcal{P}$ that reside on boundary segments or points are called **boundary nodes**. The mesh $\mathcal{M}$ must conform to the boundary $\mathcal{B}$ in the following sense: the end-points of boundary segments and points must be mesh nodes, the edges of the mesh cannot cross boundary segments, and the boundary segments must be decomposed into a set of mesh edges.

The mesh can be **structured** or **unstructured**. For a structured mesh, the local geometrical and combinatorial structure of the mesh at each of its nodes is identical, with the possible exception of the mesh's boundary nodes. Unstructured meshes allow for full flexibility, and the local geometrical

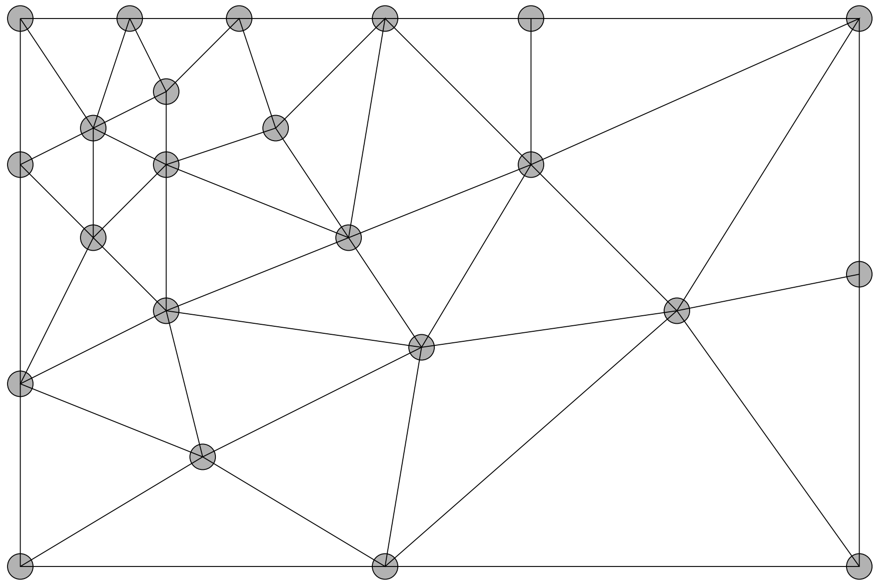


Figure 2.1: A small quasi-uniform mesh

and combinatorial structure can differ from node to node. Structured meshes often use quadrilateral elements, especially rectangles and squares. Unstructured meshes typically use triangular elements.

The following three mesh categories play an important part in the examples, discussion and proofs of this thesis:

- **grids:** structured meshes of quadrilateral elements.
- **quasi-uniform unstructured meshes:** an unstructured triangular mesh, whose elements are all of similar edge length. More formally, there exist two positive constants C and c such that for any two edges e_1, e_2 in the mesh $c|e_1| \leq |e_2| \leq C|e_1|$. See Figure 2.1.
- **general unstructured meshes:** no restrictions on the relative size of elements, or their neighborhood structure.

Given a general set of points, any of its triangulations is an example for an unstructured mesh. One well-known algorithm to construct a triangulation of a point set is the Delaunay triangulation.

Definition 2.1.1 (Delaunay triangulation) *The Delaunay triangulation of a point set $\mathcal{P}$ is a decomposition of the convex-hull of $\mathcal{P}$ into Delaunay triangles. A triangle with vertices in $\mathcal{P}$ is a Delaunay triangle if its circumscribed ball contains no points of $\mathcal{P}$ in its interior. The dual of the Delaunay triangulation is the* ***Voronoi diagram****. It assigns to each Delaunay node a cell which is the set of domain points closer to it than to any other Delaunay node. Two nodes share an edge in the Delaunay triangulation if their Voronoi cells are adjacent to each other.*

The ***constrained*** *Delaunay triangulation of $\mathcal{P}$ with respect to domain description $\mathcal{B}$ is a decomposition of $\mathcal{B}$ into triangles $\mathcal{T}$ with vertices from $\mathcal{P}$ such that the circumscribed sphere of any $T \in \mathcal{T}$ contains no points of $\mathcal{P}$ visible from the interior of T. It is assumed that the vertices of the boundary description $\mathcal{B}$ are contained in $\mathcal{P}$.*

A mesh is generated as an intermediate step of a numerical method to compute or simulate physical quantities over the original domain. Not all meshes perform equally well in the subsequent numerical computations. Numerical and discretization errors depend on the geometric shape and size of the mesh elements. The following definitions help formalize the concepts of element size and shape:

Definition 2.1.2 (Edge Length Function (el_M)) *Let M be mesh over a two dimensional domain Ω. For each $x \in \Omega$, $\mathsf{el}_M(x)$ is defined to be the length of the longest edge of all the mesh elements that contain x. (Note that only the mesh nodes and the mesh edges are contained in more than one element.)*

This definition was used in several mesh generation papers, such as [9, 64, 52].

The quality of the shape of a triangular element is measured using its aspect ratio. Generally, long and skinny triangles are considered to be undesirable, as was discussed in Section 1.2.2.

Definition 2.1.3 (aspect ratio) *The **aspect ratio** of a mesh element is the ratio R/r between the radius r of the largest ball contained in the element, and the radius R of the smallest ball containing the element. The aspect ratio of a mesh is the maximal aspect ratio of its elements.*

There are many possible definitions for aspect ratio. These definitions are equivalent in 2D in the sense that if a triangle is bounded away from being an ill-shaped triangle under one aspect ratio, it is bounded away under the others as well. Other examples for aspect ratios are:

- Radius-edge ratio: $\frac{\mathcal{R}}{\ell}$, where $\mathcal{R}$ is the radius of the circumscribed ball and L the length of the shortest edge.
- Smallest angle of the triangle.
- Largest ratio of the length of an edge to the length of height to the edge: e/h.

The equivalence between the above allows interchanging them as convenient. A mesh is said to be of good aspect ratio if its aspect ratio is smaller than a given constant C, or if its smallest angle is larger than a given constant angle θ. The Delaunay triangulation is often used in mesh generation since it maximizes the smallest angle in the triangulation; see for example [8].

The edge length function el is a measure of the mesh element size at any point; the aspect ratio is a measure of the element shape. These two qualities are often at odds: to produce a good aspect ratio mesh conforming to the boundary smaller elements must be used, thus increasing the size of the resulting mesh and consequently the work performed by the numerical method.

2.2 Spacing functions and mesh functions

In this section the concept of spacing functions is formalized. Spacing functions are slowly changing functions that describe the typical distance between points in a well-spaced point set. Since the typical edge length of a good aspect ratio mesh varies slowly across the domain, slowly changing functions play an important role in mesh generation [64].

Definition 2.2.1 (1-Lipschitz) *A function f is 1-Lipschitz over a domain Ω if for any two points x,y in Ω, $|f(x) - f(y)| \leq ||x - y||$.*

For the point set to be well-spaced according to 1-Lipschitz spacing function, points should not get too close. One way to enforce this is by drawing a ball around each node and requiring these balls to be disjoint. The radius of the balls is proportional to the spacing function at each node. Since balls play such a pivotal role, the following short-hand notation is convenient:

Definition 2.2.2 $\mathsf{B}(x,r)$ *is the ball of radius* r *centered at* x*.*

Definition 2.2.3 (β-spaced point set) *Let* $\beta > 1$ *be a real number. A point set* $\mathcal{P}$ *is* ***β-spaced*** *according to a* 1*-Lipschitz function* f *if for any two points* $p, q \in \mathcal{P}$, $\frac{f(p_1)}{\beta} + \frac{f(p_2)}{\beta} < \|p-q\|$*. The function* f *is then referred to as the* β*-spacing function of* $\mathcal{P}$*. Note that the balls* $\mathsf{B}(p, f(p)/\beta)$ *and* $\mathsf{B}(q, f(q)/\beta)$ *are required to be disjoint. These balls will be referred to as the* f*-balls, or when there is no need to explicitly name* f*, the point-balls.*

Remark: Using a β-spacing function which is 1-Lipschitz is equivalent to using a 1-spacing function which is $\frac{1}{\beta}$-Lipschitz. Here the former notation is preferred, and all the spacing functions are normalized to be 1-Lipschitz.

Given a point-set $\mathcal{P}$, the following is a 1-Lipschitz function that 2-spaces $\mathcal{P}$:

Definition 2.2.4 *The* ***nearest neighbor*** (NN) *function of a point set* $\mathcal{P} \subset \Omega$ *assigns to each point* $p \in \mathcal{P}$ *the distance to the point* $q \in \mathcal{P}$ *nearest to it. It can be extended to the domain* Ω *by assigning to a point* $x \in \Omega$ *the radius of the smallest closed ball centered at* x *and containing at least two points from* $\mathcal{P}$*.*

Therefore, every point-set is 2-spaced by a slowly changing spacing function. This implies that spacing functions in themselves are not strong enough to capture all the qualities of well-spaced points. Intuitively, spacing functions only prevent points from approaching each other, but not from straying too far away. Section 2.3 below strengthens the definition of a point set spaced by a function so that the strengthened definition can also capture the idea that gaps are not allowed to form.

2.2.1 Previous uses of mesh functions

There are previous instances of associating local spacing or local scale functions with meshes:

- **In mesh generation:** algorithms that construct the mesh using *advancing front* methods use a function describing the typical element size as a function of location. In advancing front methods the mesh is "grown" element by element. A new element, whose size is determined by the element size function, is glued to the "front" of the region of old elements. The front is typically initiated at the boundaries. Mavriplis [47] surveys a variety of advancing front techniques. He classifies size functions as explicit or implicit. In implicit functions, the size is inferred from the boundary discretization. A popular way to represent explicit functions is using **background grids**, with function values at the grid nodes interpolated to interior nodes. The background grid can be unstructured [58, 35], Cartesian [59], or quadtree based [45].

- **In mesh size analysis:** Mitchell and Vavasis [52] and Ruppert [64] use functions describing the typical local size to bound the number of elements in the meshes their algorithms produce. Mitchell and Vavasis use the **characteristic edge length function** to show that any tetrahedralization of the same domain has characteristic edge length function comparable via a constant factor to the octree based tetrahedralization of their algorithm, and consequently also a similar number of tetrahedra. Ruppert used a similar function, the **edge length function**, in addition to a function called the **local feature size**. He showed that all good aspect ratio meshes of the same domain have edge length functions larger than the local feature size function by a constant factor at most. Whereas Mitchell and Vavasis compared the sizes of good aspect ratio meshes relative to each other, Ruppert, via the local feature size function, provided an absolute way to measure (up to a constant factor) the size of the smallest mesh.

These previous instances, in which mesh functions were used to analyze the mesh or to generate the mesh, motivated this work. Ruppert's local feature size function was especially influential, and the following definitions and results by Ruppert are used in the text:

Definition 2.2.5 (Ruppert: Local feature size (lfs)) *Given a PSLG $\mathcal{B}$, the local feature size at point x, $\mathsf{lfs}_{\mathcal{B}}(x)$, is the radius of the smallest ball centered at x that intersects two non-incident vertices or segments of G.*

Note that the lfs function depends only on the boundary $\mathcal{B}$ and not on the actual discretization of the domain. Also, for more complex domains this definition by Ruppert should be amended by using the geodesic distance rather than the Euclidean distance.

Definition 2.2.6 (Geodesic distance) *Given a domain Ω, the geodesic distance between two points $x, y \in \Omega$ is the length of the shortest path between them that is wholly contained in Ω.*

The local feature size at the point is then amended to be the geodesic distance to the furthest of the two nearest non-incident features. Without this definition, domains that curve back upon themselves can introduce unnecessarily small feature sizes.

Lemma 2.2.7 (Ruppert) *The lfs function is 1-Lipschitz.*

Proof: Let x and y be two domain points. Without loss of generality, let $\mathsf{lfs}(y) > \mathsf{lfs}(x)$. The ball $\mathsf{B}(x, \mathsf{lfs}(x))$ contains two non-incident input features, hence the ball centered at y of radius $\mathsf{lfs}(x) + \|x - y\|$ contains them as well, and

$$\mathsf{lfs}(y) - \mathsf{lfs}(x) \leq \mathsf{lfs}(x) + \|x - y\| - \mathsf{lfs}(x) = \|x - y\|.$$

□

Under certain restrictions on the boundary $\mathcal{B}$ (e.g., the smallest domain angle between segments of $\mathcal{B}$ has to be larger than or equal to 90°), Ruppert showed the following theorem:

Theorem 2.2.8 (Ruppert) *Given a PSLG $\mathcal{B}$, and $0 < \theta \leq 20°$, there exists a constant C depending only on θ such that for any triangulation T of $\mathcal{B}$ with minimum angle bound θ*

$$\mathsf{el}_T(x) \leq C\mathsf{lfs}_{\mathcal{B}}(x), \qquad x \in \Omega.$$

Relationship between lfs and NN: $\mathsf{lfs}_{\mathcal{B}}$ depends on the domain description, whereas $\mathsf{NN}_{\mathcal{P}}$ depends on the point set of an actual discretization of the domain. Note that $\mathsf{NN}_{\mathcal{P}}$ can be much smaller than $\mathsf{lfs}_{\mathcal{B}}$, but Theorem 2.2.8 implies it can be at most a constant factor larger for a good aspect ratio mesh.

Therefore, for $\mathsf{NN}_{\mathcal{P}}$, a stronger result can be proven (compare to Theorem 2.2.8, under the same assumptions):

Theorem 2.2.9 *Let $M = (\mathcal{P}, E, \mathcal{B})$ be a mesh with smallest angle θ. There exist two positive constants C_1, C_2 depending on θ only such that*

$$C_1 \mathsf{el}_M(x) \leq \mathsf{NN}_{\mathcal{P}}(x) \leq C_2 \mathsf{el}_M(x).$$

Proof: Given the mesh M, define a new domain description $\mathcal{B}' = \mathcal{P}$. That is, the domain $\mathcal{B}'$ is composed of the mesh nodes. Theorem 2.2.8 then implies that $C_1 \mathsf{el}_M(x) \leq \mathsf{lfs}_{\mathcal{B}'} = \mathsf{NN}_{\mathcal{P}}(x)$. The other direction is also simple: First look at the inequality for $p \in \mathcal{P}$: It is obvious that $\mathsf{NN}_{\mathcal{P}}(p) \leq \mathsf{el}_M(p)$ since the longest edge at p must be at least as long as its distance from its nearest neighbor. Now look at $x \in \Omega$, and suppose x is in triangle T. Let p be one of the vertices of T. Since M is a good aspect ratio mesh, there exists a constant C such that $\mathsf{el}(p) \leq C\mathsf{el}(x)$ (since the edges of neighboring triangles can differ by at most a constant factor). The distance between x and p is at most $\mathsf{el}(x)$, therefore:

$$\mathsf{NN}_{\mathcal{P}}(x) \leq \mathsf{NN}_{\mathcal{P}}(p) + \mathsf{el}(x) \leq \mathsf{el}(p) + \mathsf{el}(x) \leq (C+1)\mathsf{el}(x).$$

□

2.3 Well-spaced points

Points can be spaced according to a spacing function, and yet not be well-spaced. The problem is that Definition 2.2.3 merely prevents the points from getting too close, but gaps can still form. In this section the concept of *gaps* and ways to control gaps are developed. At the end of this section we are finally able to formally define well-spaced points.

Definition 2.3.1 (gap) *Let $\mathcal{L}$ be a positive constant. An **$\mathcal{L}$-gap** at point x is a ball $B = \mathsf{B}(c, \mathcal{L}f(x))$ such that x is on the circumference of B. $\mathcal{L}$ is called the **gap parameter**. The $\mathcal{L}$-gap B at point x is said to be empty if $B \cap \mathcal{P} = \emptyset$.*

Restrictions on gaps should be carefully designed. Clearly, a point set or mesh of a domain can have arbitrarily large gaps that are mostly outside of the domain, see the gap at point y in Figure 2.2. Nonetheless, it is crucial to control gaps that are near the boundaries and are not completely contained in the domain, as well as gaps that are contained in the domain. The following definition controls gaps in a manner that takes into consideration the domain boundaries:

Definition 2.3.2 (Bounded gaps property) *Let $\mathcal{L}$ be a positive constant. Consider the following two types of gaps:*

- *A gap $B = \mathsf{B}(c, \mathcal{L}f(x))$ passing through x is an **interior $\mathcal{L}$-gap** at x if the radial from x to c is wholly within the domain. For example, the gap at point x in Figure 2.2 is an interior gap.*
- *A gap $B = \mathsf{B}(c, \mathcal{L}f(x))$ is a **boundary $\mathcal{L}$-gap**, if there exists a boundary segment S of the mesh such that both $c, x \in S$ and the radial from x to c is in S. This is a one-dimensional version of the gap notion, with the intent to pose restrictions on $B \cap S$ rather than B, i.e. a boundary gap is called empty if $B \cap S \cap \mathcal{P}$ is empty. For example, the gap at point z in Figure 2.2 is a boundary gap.*

A mesh $M = (\mathcal{P}, E, \mathcal{B})$ or a point set $\mathcal{P}$ is said to have the bounded gaps property with gap parameter $\mathcal{L}$, or the bounded $\mathcal{L}$-gaps property, if its empty gaps are all of gap parameter smaller than or equal to $\mathcal{L}$.

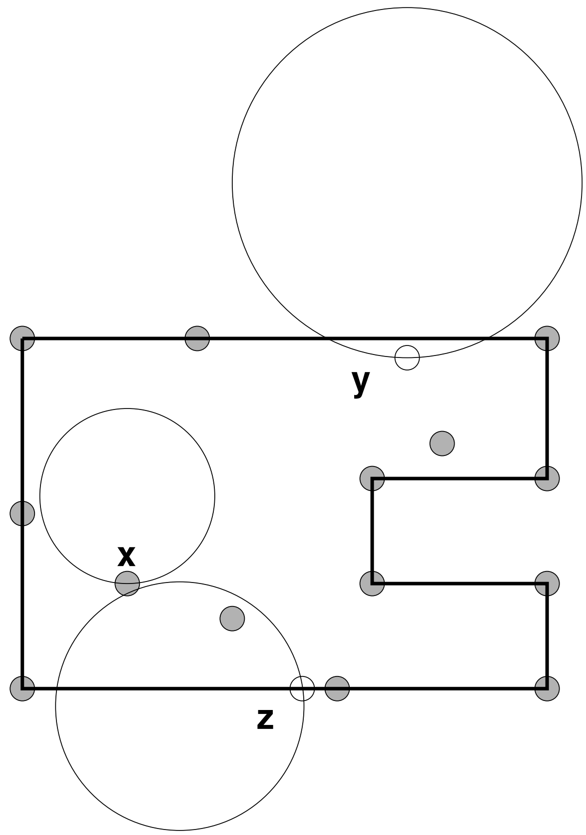


Figure 2.2: A domain with a set of points $\mathcal{P}$. The points of $\mathcal{P}$ are shaded, whereas an arbitrary domain point, such as point y or z, is unshaded. Several types of gaps are shown. Note that an empty gap can be located at an arbitrary point and contains no points of $\mathcal{P}$. The gap at x is an empty interior gap. The empty circle at point y is not a gap since its center is outside the domain. The gap at z is an empty boundary gap: it is centered on the boundary, and contains no boundary points though it does contain an interior point of $\mathcal{P}$.

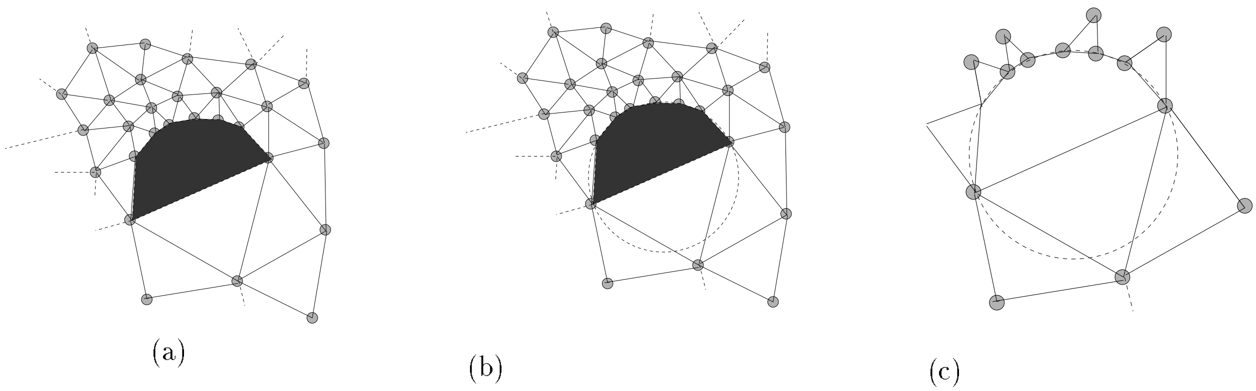


Figure 2.3: An example where the number of triangles intersecting an empty interior gap can be unbounded:(a) a domain with a (shaded) polygonal hole resembling half a circle; (b) an empty gap whose center is within the domain; (c) the triangles intersecting the gap form several connected components. This example can be generalized so that the number of connected components and the number of triangles intersecting the gap is as large as desired.

Definition 2.3.3 (Well-spaced points) *A set of points $\mathcal{P}$ in a domain $\mathcal{B}$ is* ***well-spaced*** *if there exists a* 1*-Lipschitz function f, and two positive constants β and $\mathcal{L}$ such that:*

1. *$\mathcal{P}$ is β-spaced by f over $\mathcal{B}$.*
2. *$\mathcal{P}$ has the bounded $\mathcal{L}$-gaps property with respect to f over $\mathcal{B}$.*

2.4 From a mesh to well-spaced points

This section shows that the point set of a good aspect ratio mesh is well-spaced:

Theorem 2.4.1 *Let $M = (\mathcal{P}, E, \mathcal{B})$ be a bounded aspect ratio mesh. There exists a spacing function f* 2*-spacing the mesh, such that for some positive constants $\mathcal{L}$ and b:*

1. *M has the bounded $\mathcal{L}$-gaps property with respect to f.*
2. *The geodesic distance between a point $p \in \mathcal{P}$ and a boundary segment S, such that p is not incident to S, is greater then $bf(p)$.*

That is, $\mathcal{P}$ is well-spaced with respect to f, and interior nodes do not approach the boundary too closely with respect to f.

Proof: See the following lemmas in this section.

Lemma 2.4.2 *Let $M = (\mathcal{P}, E, \mathcal{B})$ be a mesh. $\mathcal{P}$ is* 2*-spaced according to $\mathsf{NN}_{\mathcal{P}}$.*

Proof: Let $p, q \in \mathcal{P}$. $\|p - q\| \geq 0.5(\mathsf{NN}(p) + \mathsf{NN}(q))$. □

To show that a good aspect ratio mesh possesses the bounded gaps property, the size of its empty gaps must be bounded in terms of the spacing function NN. One possible way to obtain this result is to show that the number of mesh triangles intersecting an empty gap is bounded. Unfortunately, this is not always true. A counter-example is displayed in Figure 2.3.

One characteristic of this counter-example is that the triangles intersecting the gap form more than one connected component. Two triangles belong to the same *connected component* with respect to the gap if there is a path between them that is wholly contained in the gap. For the purpose of the discussion here, disallow paths going through the triangle nodes themselves so that two triangles attached only at a triangle node are not considered connected. Note that each connected component has a bounded number of triangles. It suffices to look at the number of triangles in a connected component in order to show the bounded gaps result.

Lemma 2.4.3 *Let $M = (\mathcal{P}, E, \mathcal{B})$ be a mesh with smallest angle θ. Let B be a ball. The intersection of B and M can be decomposed into connected components of triangles. Let N be one of the connected components. If $N \cap B \cap \mathcal{P} = \emptyset$ then N contains at most $2 + \frac{12}{\sin^4\theta}$ triangles of M.*

Proof: without loss of generality let the radius of B be of unit length. Let $N_i \subseteq N$ be the set of triangles intersecting B with i edges, $i = 1, 2, 3$.

First, it is shown that $|N_1| \leq 2|N_2| + 2|N_3| + 2$, and hence that $|N| \leq 3|N_2 \cup N_3| + 2$. Each $T_1 \in N_1$ intersects B with an edge e. If e is not incident on any other triangle then $N = \{T\}$ and for this special case the proof is done. Otherwise, let $T \in N$ be the other triangle incident on e. If $T \in N_1$ then $N_1 = \{T, T_1\}$ and $N_2 = N_3 = \emptyset$. Otherwise, $T \in N_2 \cup N_3$. Each $T \in N_2$ can have at most two neighbors from N_1. Each $T \in N_3$ can have at most two neighbors from T_1 unless it is the special case $N_1 = \{T_1, T_2, T_3\}$, $N_2 = \emptyset$ and $N_3 = \{T\}$.

Now, a volume argument is used to count N_3. Any triangle T in N_2 can be reduced to a smaller type N_3 triangle T', by a similarity transformation to a triangle nested in T. This only decreases the volume the triangle occupies. Hence, it suffices to count N_3. Let $T \in N_3$. Using another similarity transform, shrink T to a smaller triangle nested in T, with two vertices a, b on the circumference of B. Referring to Figure 2.4, look at a chord starting at a, such that the chord separates the triangle from the center. (There is only one triangle that contains the center). Using the fact that angle $\alpha \geq \theta$, and that $\psi \leq 90° - \theta$, the chord is of length at least $L = 2\sin\theta$ and H, the vertical line from the middle of the chord to point h is at least $\sin^2\theta$. Therefore, a circle of radius $0.5\sin^2\theta$ is contained in $T \cap B$, and $|N_2 \cup N_3| \leq \frac{4}{\sin^4\theta}$, or $|N| \leq 2 + 3\frac{4}{\sin^4\theta}$. □

The following theorem shows that a good aspect ratio mesh can not have large empty interior gaps in terms of $\mathsf{NN}_{\mathcal{P}}$:

Theorem 2.4.4 *Let $M = (\mathcal{P}, E, \mathcal{B})$ be a mesh with smallest angle θ. There exists a constant $\mathcal{L}$, depending only on θ, such that for any $\mathcal{L}_1 \geq \mathcal{L}$ all interior $\mathcal{L}_1$-gaps with respect to $\mathsf{NN}_{\mathcal{P}}$ must contain a node of $\mathcal{P}$.*

Proof: Let $x \in \Omega$ be an arbitrary point, B a ball through x with radius r and center c, such that the radial from x to c is contained in Ω. Look at N, the connected component of triangles that intersect the radial. Assume B contains no node of N, then by Lemma 2.4.3 N contains at most $C(\theta)$ triangles of M, where C is a constant depending on θ. The center c is in some triangle $T = (k, l, m)$. Let $(k, l) = \ell$ be the longest edge of T. Clearly, $r \leq |c - k| \leq \ell$, and therefore $\mathsf{el}(c) \geq r$. x is in some triangle T' which is at most $C(\theta)$ triangles from T, and the longest edge of neighboring triangles in a good aspect ratio mesh differ by a constant factor $E(\theta)$ at most. Therefore, setting $D(\theta) = E(\theta)^{C(\theta)}$, $\mathsf{el}(x) \geq D(\theta)\mathsf{el}(c) \geq D(\theta)r$. By Theorem 2.2.9, $\mathsf{NN}_{\mathcal{P}}(x) \geq C_1\mathsf{el}(x) \geq D(\theta)r$, and the radius of an empty ball at x is bounded in terms of $\mathsf{NN}_{\mathcal{P}}$. □

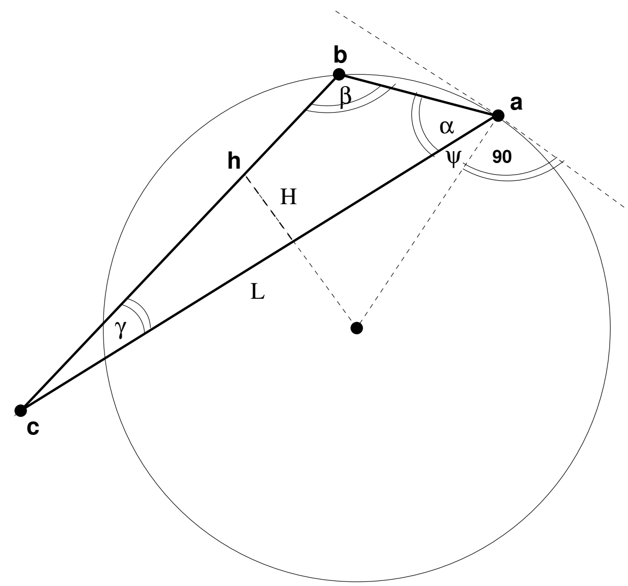


Figure 2.4: A triangle intersecting the ball with three edges.

The following theorem shows that a good aspect ratio mesh can not have large empty boundary gaps in terms of $\mathsf{NN}_{\mathcal{P}}$:

Theorem 2.4.5 *Let $M = (\mathcal{P}, E, \mathcal{B})$ be a mesh with smallest angle θ. There exists a constant $\mathcal{L}$ depending on θ only such that all boundary $\mathcal{L}$-gaps with respect to $\mathsf{NN}_{\mathcal{P}}$ contain a boundary node of $\mathcal{P}$.*

Proof: Let x be a point on a boundary segment S. Let $p, q \in S \cap \mathcal{P}$ be the end-points of the boundary edge of S that contains x. By definition of the el function, the size of the gap at x that contains either p or q must be smaller than $\mathsf{el}(x)$. By Theorem 2.2.9, the gap size is at most $\frac{1}{C_1}\mathsf{NN}_{\mathcal{P}}(x)$. Set $\mathcal{L} > \frac{1}{C_1}$.

□

Theorem 2.4.6 *The geodesic distance between a node p of mesh $M = (\mathcal{P}, E, \mathcal{B})$ and a boundary segment S, such that p is not incident to S, is at least $b\mathsf{NN}_{\mathcal{P}}(p)$, where $b = \sin\theta$.*

Proof: The geodesic distance between p and S must be larger than the distance between p and e, where e is the closest opposite edge to p among the triangles T incident to p. Let H be the height to e, let L be the length of e_1, one of the other edges of T. Since the angle between e_1 and e is larger than θ, $H > L\sin\theta > NN_{\mathcal{P}}(p)\sin\theta$. □

2.5 From well-spaced points to a mesh

In this subsection, well-spaced points are shown to be strong enough to produce a good aspect ratio mesh, if points are not too close to a boundary segment they are not incident to. Furthermore, the resulting mesh has el function similar, up to a constant factor, to the original spacing function used.

Theorem 2.5.1 *Let $\mathcal{P}$ be a well-spaced point set with respect to a 1-Lipschitz function f, i.e. there exists positive constants β and $\mathcal{L}$ such that:*

I. $\mathcal{P}$ is β-spaced by f.

II. $\mathcal{P}$ has the bounded $\mathcal{L}$-gaps property with respect to f.

Furthermore, assume there exists a positive constant b such that:

III. The geodesic distance between any point $p \in \mathcal{P}$ and a boundary segment S not incident to p is greater than $bf(p)$.

Then, letting M be the constrained Delaunay triangulation of $\mathcal{P}$:

1. *The smallest angle θ of a triangle whose circumcircle is an interior gap is bounded by:*

$$\sin\theta \geq \frac{4}{\mathcal{L}(2+\beta)}$$

2. *The smallest angle θ of a triangle whose circumcircle is not an interior gap is bounded by:*

$$\sin\theta \geq \frac{4}{2\mathcal{L}^2(1+1/b)(2+\beta)}$$

Proof: Let DT($\mathcal{P}$) stand for the constrained triangulation of $\mathcal{P}$ with respect to the boundary $\mathcal{B}$. Let $T \in$ DT($\mathcal{P}$) be a constrained Delaunay triangle with vertices p, q and r. Let (p,q) be the shortest edge of T, $\|p-q\| = \ell$. Since $\mathcal{P}$ is β-spaced, $f(p)+f(q) \leq \beta\ell$. Without loss of generality, $f(p) \leq \beta\ell/2$, and $f(q) \leq f(p)+\ell < \ell(1+\beta/2)$. The Delaunay ball B of T is of radius R. Furthermore, it is empty in the constrained sense, i.e. no other node of $\mathcal{P}$ is visible to p, q or r within B. Let c be the center of B. If c is visible to either p or q, B is an interior gap, otherwise B contains a boundary gap. We argue by cases:

1. B is an interior gap: without loss of generality, c is visible to q. By property II, the radius R of B must be smaller than $\mathcal{L}f(q)$. Hence, $R \leq \ell\mathcal{L}(1+\beta/2)$. The triangle then has radius-edge ratio $R/\ell \leq \mathcal{L}(1+\beta/2)$, and the corresponding angle bound

$$\sin\theta \geq \frac{4}{\mathcal{L}(2+\beta)}$$

2. B is not an interior gap: There exists a boundary segment S blocking c from both p and q. Let S be the first such segment intersecting the radial from q to c. Note that the segment S can not be the edge (p,q) itself, since this would imply that the shortest edge separates the center from the triangle, and consequently that the smallest triangle angle is larger than 90° — a contradiction. Also, the vertices of S are on the circumference of or outside B, otherwise B would contain nodes in its interior visible to p or q and can not be a Delaunay ball; these nodes are either S's vertices, or the vertices of a segment hiding S's vertices or, in turn, a segment hiding those and so on recursively. As the number of segments is finite, the chain stops at some nodes which are then visible to p or q.

   Without loss of generality, let q be the node not incident to S. By property III, the distance ℓ between q and S is at least $bf(q)$. Also, the distance ℓ must be smaller than $\mathcal{L}f(q)$, otherwise

B contains an interior $\mathcal{L}$-gap through q and as by the bounded gaps property must contain a node visible to q, a contradiction to the fact B is a constrained Delaunay circumsphere. Hence

$$bf(q) < \ell < \mathcal{L}f(q)$$

We now argue that if the radius R of B is greater than $(\mathcal{L}/2 + 2\mathcal{L}^2)f(q)$ then B contains the vertices of S. As before, this would imply that some node in B is visible to q, a contradiction to the fact B is a constrained Delaunay circumsphere. We outline the argument before going into the details. We proceed by showing that S can not be too big; it is close to q so its length must be bounded by some function of $f(q)$, since f changes slowly and all the boundary gaps on S are bounded. Then, we use this fact to conclude that R must be small or it would contain the vertices of S.

Let y be the point at the intersection of S and the radial between q and c. $f(y) < f(q) + \ell$. The boundary gap at y is therefore at most $\mathcal{L}(f(q) + \ell)$. This is the maximal distance between y and any of the vertices of S. Also note that we may assume that S is a chord through y with respect to q. The chord is the shortest segment through y, so if the chord and its vertices are shown to be inside B, then the same is true for any other segment of the same size or larger through y.

To argue that B contains S, it suffices to argue that B would contain S's vertices for the cases y is at distance $\ell = bf(q)$ and $\ell = \mathcal{L}f(q)$, the closest and the furthest y can be from q. The rest of the locations form a linear interpolation between the two. At distance $\ell = bf(q)$, the length of the chord is at most $\mathcal{L}f(q)(b+1)$. R would contain S if

$$\frac{R}{f(q)} > \frac{b}{2} + \frac{(1+b)^2\mathcal{L}^2}{2b},$$

or, since b can be assumed to be smaller than one:

$$\frac{R}{f(q)} > \frac{1}{2} + \frac{2\mathcal{L}^2}{b}.$$

At distance $\ell = \mathcal{L}f(q)$, the length of the chord is at most $\mathcal{L}f(q)\mathcal{L}(\mathcal{L}+1)$. R would contain S if

$$\frac{R}{f(q)} > \frac{\mathcal{L}}{2} + \frac{(1+\mathcal{L})^2\mathcal{L}}{2}$$

Therefore, taking the two restrictions together, and assuming $\mathcal{L} > 1$, R must obey

$$\frac{R}{f(q)} < 2\mathcal{L}^2(1 + \frac{1}{b})$$

or B would contain S. The bound on θ follows.

□

The following theorem connects the function f β-spacing $\mathcal{P}$ to the NN function of $\mathcal{P}$, and by Theorem 2.2.9, the connection between $\mathsf{el}_{\mathrm{DT}(p)}$ and f also follows.

Theorem 2.5.2 *Let $\mathcal{P}$ be a set of points β-spaced by a 1-Lipschitz function f, with the bounded gap property parameterized by $\mathcal{L}$, then:*

$$\frac{\mathsf{NN}_{\mathcal{P}}(x)}{\mathcal{L}(1+\mathcal{L})(2+\mathcal{L}+\mathcal{L}^2)} \leq f(x) \leq (2\beta+1)\mathsf{NN}_{\mathcal{P}}(x)$$

Proof:

- $f(x) \leq (2\beta+1)\mathsf{NN}(x)$: First assume $p \in \mathcal{P}$, and let $q \in \mathcal{P}$ be the $\mathcal{P}$ point nearest to p. Then $\mathsf{NN}(p) = \|p-q\|$. On the other hand, since $\mathcal{P}$ is β-spaced by f, $f(p) \leq \beta\|p-q\|$. Therefore, $f(p) \leq \beta\mathsf{NN}(p)$.

  Now let $x \in \Omega$. By definition, $\mathsf{NN}_{\mathcal{P}}(x)$ is the radius r of the smallest ball that contains two (visible from q) points, say p and q, from $\mathcal{P}$. Let $\|x-p\| = r$ and $\|x-q\| = r_1 \leq r$. Because f is 1-Lipschitz, $f(x) \leq f(p) + r \leq \beta\mathsf{NN}(p) + r$. But $\mathsf{NN}(p) \leq 2r$, for q is no more than $2r$ (geodesic) distance away from p. Thus, $f(x) \leq 2\beta r + r \leq (2\beta+1)\mathsf{NN}(x)$.

- $\mathsf{NN}(x)/(\mathcal{L}(1+\mathcal{L})(2+\mathcal{L}+\mathcal{L}^2)) \leq f(x)$: Let $x \in \Omega$. Look at a ball of radius $\mathcal{L}f(x)$ through x. If the center of B is visible from x, then by the the bounded gaps property it contains a point of $\mathcal{P}$ visible to x. Otherwise, a segment S blocks the center of B. One of the end-points of S, p, is then at distance at most $f(x)\mathcal{L}(\mathcal{L}+1)$. This last step used the bounded gaps at the boundary S property. Similar argument for p leads to a point q at distance $f(p)(\mathcal{L}(1+\mathcal{L})(2+\mathcal{L}+\mathcal{L}^2))$ from p. Therefore, a circle of radius $f(x)$ around x contains two points p and q.

□

2.6 Summary

This Chapter introduced the concept of well-spaced points, and proved that it is equivalent to the concept of a bounded aspect ratio mesh in two dimensions. In particular, we have shown:

- Every bounded aspect ratio mesh is composed of well-spaced points. Furthermore, interior domain points are not too close to the boundary relative to the local spacing,

- The constrained Delaunay triangulation of a well-spaced point set, where no point of the set is too close to a non-incident boundary segment relative to the local spacing, is a bounded aspect ratio mesh conforming to the domain.

This equivalence lays the foundations to the rest of this thesis. It enables us to develop algorithms for mesh coarsening and mesh generation that focus on the underlying well-spaced point set of the mesh.

Chapter 3

Well-Spaced Points in Three Dimensions

This chapter discusses the generalization of well-spaced points to three dimensions, and shows that the point sets of the following classes of tetrahedral meshes are well-spaced:

- Bounded aspect ratio tetrahedral meshes
- Bounded radius-edge ratio tetrahedral Delaunay meshes

The radius-edge ratio is a new aspect ratio introduced in this chapter, and is weaker than the standard aspect ratio in the sense that a bounded aspect ratio tetrahedron is also of bounded radius-edge ratio but not conversely. The radius-edge ratio is important both since the control volume method obtains optimal rates of convergence when solving Poisson's equation over a bounded radius-edge ratio mesh (see Section 1.2.3), and since this ratio strongly characterizes Delaunay meshes of well-spaced points.

Recall that in two dimensions, well-spaced points are equivalent to bounded aspect ratio meshes in the sense that the point set of such a mesh is well-spaced, and its nodes stay away from boundary elements they are not incident to, and conversely, the constrained Delaunay triangulation of such point sets is a bounded aspect ratio mesh.

It is important to note that in order to obtain this two dimensional equivalence the well-spacing of the points must be augmented by restrictions on the interaction of the nodes with the boundary. Without such restrictions, a node might approach a boundary segment too closely and force a bad aspect ratio triangle. Clearly, such additional conditions will be necessary in three dimensions as well.

Unfortunately, generalizing the results of the previous chapter to three dimensions is not straight forward even when the interaction with the boundary is put aside, for example when the domain is very simple or infinite.

In three dimensions, it is still the case that the nodes of any bounded aspect ratio mesh are well-spaced. The problem is in showing the other direction, the direction of tetrahedralizing well-spaced points to form a good aspect ratio mesh. It is reasonable to conjecture that every well-spaced point set can be tetrahedralized to form a good aspect ratio mesh. However, the Delaunay tetrahedralization fails to oblige us in that respect. Currently, it is unknown if such a tetrahedralization algorithm exists, or if there exists a counter-example: a well-spaced point set with the property that none of its tetrahedralizations is of bounded aspect ratio.

This motivates introducing the class of bounded radius-edge ratio Delaunay tetrahedralization meshes. Ignoring the boundaries, this class allows for the following equivalence that is similar to the two dimensional equivalence: the point set of bounded radius-edge ratio Delaunay mesh is well-spaced, and the Delaunay tetrahedralization of well-spaced points is a bounded radius-edge ratio mesh.

Furthermore, the results in this chapter show that a Delaunay mesh of bounded radius-edge ratio shares some of the geometrical and combinatorial properties of a bounded aspect ratio mesh. The reason for this similarity can be intuitively understood by considering the structure of the Voronoi diagram of a bounded radius-edge ratio mesh. This chapter shows the Voronoi diagram is composed of well-shaped (good aspect ratio) convex polytopes, and hence very similar to the well-shaped simplicial decomposition of a bounded aspect ratio mesh.

Section 3.1 defines well-spaced points in three dimensions, and Section 3.2 shows that the nodes of a bounded aspect ratio mesh in 3D is a well-spaced point set. The rest of the sections build towards showing that the same is true for a bounded radius-edge ratio Delaunay mesh: Section 3.3 defines the radius-edge ratio, Section 3.4 shows that the 1D skeleton of such a mesh is a density graph, Section 3.5 discusses the implications of the density graph result to the shape of the Voronoi diagram and finally Section 3.6 shows that the nodes of a bounded radius-edge ratio mesh are well-spaced.

3.1 Well-spaced points in 3D

A point set is well-spaced, in three dimensions as well as in two dimensions, if points do not approach each other too closely and if there are no large gaps:

Definition 3.1.1 (Well-spaced points) *A set of points $\mathcal{P}$ in a domain $\mathcal{B}$ is* ***well-spaced*** *if there exists a 1-Lipschitz function f, and two positive constants β and $\mathcal{L}$ such that:*

1. *$\mathcal{P}$ is β-spaced by f over $\mathcal{B}$.*

2. *$\mathcal{P}$ has the bounded $\mathcal{L}$-gaps property with respect to f over $\mathcal{B}$.*

Only the definition of bounded gaps needs to be generalized to three dimensions (see Definition 2.3.1 for the two dimensional definition):

Definition 3.1.2 (Bounded gaps in 3D) *Let $\mathcal{L}$ be a positive constant. Consider the following three types of gaps:*

- *A gap $B = \mathsf{B}(c, \mathcal{L}f(x))$ passing through x is an* ***interior $\mathcal{L}$-gap*** *at x if the radial from x to c is wholly within the domain.*

- *A gap $B = \mathsf{B}(c, \mathcal{L}f(x)) \cap S$ is a* ***boundary-segment $\mathcal{L}$-gap****, if there exists a boundary segment S of the mesh such that the radial from x to c is wholly in S.*

- *A gap $B = \mathsf{B}(c, \mathcal{L}f(x)) \cap F$ is a* ***boundary-face $\mathcal{L}$-gap****, if there exists a boundary face F of the mesh such that the radial from x to c is wholly in F.*

A mesh $M = (\mathcal{P}, E, \mathcal{B})$, or a point set $\mathcal{P}$, is said to have the bounded-gap property with gap parameter $\mathcal{L}$ if its empty gaps are all of gap parameter smaller than $\mathcal{L}$.

3.2 Bounded aspect ratio meshes have well-spaced nodes

The following lemma and theorem show that the gaps of a bounded aspect ratio mesh in three dimensions are bounded with respect to the function $\mathsf{NN}_{\mathcal{P}}$. Since every point set $\mathcal{P}$ is spaced by its $\mathsf{NN}_{\mathcal{P}}$ function, this implies the mesh nodes are well-spaced with respect to $\mathsf{NN}_{\mathcal{P}}$.

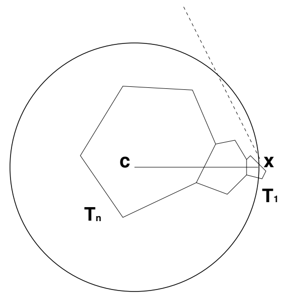


Figure 3.1: Illustration for Lemma 3.2.1. The Voronoi cells are well-shaped so if the first cell T_1 is small, the neighboring cells must remain below the dashed lines. This implies some Voronoi cells are fully contained in the ball. Consequently, some Delaunay node is in the ball, and the ball is not empty.

Lemma 3.2.1 *Let $M = (\mathcal{P}, E, \mathcal{B})$ be a bounded aspect ratio tetrahedral mesh, i.e., the smallest dihedral angle of the tetrahedra of M is larger than θ. Let B be a an interior gap with center c. Let x be a point on the circumference of B such that the radial $(x, c) \in \Omega$.*

If B does not contain any of the nodes $\mathcal{P}$, then the number of tetrahedra intersecting the radial (x, c) is bounded by a constant depending only on θ.

Proof sketch: Look at the tetrahedra $T_1, ..., T_n$ that intersect the radial, where $x \in T_1$ and $c \in T_n$. Assume that the distance between x and c is of unit length. The proof proceeds by showing that for some ϵ depending only on θ, a tetrahedron whose largest edge is smaller than ϵ can not intersect the radial. Since only large tetrahedra intersect the radial, all of them of bounded radius-edge ratio, a volume argument can then be used to show show that only a constant number of tetrahedra intersect the radial. This constant depends on θ since the volume of the tetrahedra, as well as ϵ itself, depend on θ.

Clearly, for any ϵ, if the distance of T_i from the boundaries of B is larger then ϵ then the largest edge of T_i is larger than ϵ, otherwise the gap B contains a node of T_i and is not empty. Assume by way of contradiction that triangle T_1 has largest edge smaller than ϵ. The edge length function is Lipschitz for some Lipschitz constant that depends on θ, hence $T_2, ..., T_n$ must all be below the dashed line in Figure 3.1. The slope of the dashed line depends on θ and must be smaller than $90°$. Clearly, if ϵ is set small enough, some tetrahedron T_i intersecting the radial is fully contained in the gap B, a contradiction. Therefore, the largest edge of T_1 must be larger then some constant ϵ_1. All other tetrahedra are far enough from the boundary or close neighbors of T_1 and hence can not be small either. $\square$

Theorem 3.2.2 *Let $M = (\mathcal{P}, E, \mathcal{B})$ be a bounded aspect ratio tetrahedral mesh, Let $\mathsf{NN}_{\mathcal{P}}$ be the nearest neighbor function. (Definition 2.2.4). There exist constants $0 < \mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$ such that with respect to the function $\mathsf{NN}_{\mathcal{P}}$:*

1. *edge boundary gaps are bounded by $\mathcal{L}_1$,*
2. *face boundary gaps are bounded by $\mathcal{L}_2$, and*
3. *interior gaps are bounded by $\mathcal{L}_3$.*

i.e. $\mathcal{P}$ has the bounded gaps property with parameter $\mathcal{L} = \max(\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3)$.

Proof:

1. Similar to Theorem 2.4.5.

2. Let F be a boundary face. M restricted to F is a good aspect triangulation on F, and hence the 2D corresponding theorem 2.4.4 can be used to derive the result.

3. Using Lemma 3.2.1. Let $x \in \Omega$ be an arbitrary point, B an interior gap through x. Let c be the center of B, R its radius. By Lemma 3.2.1, $\mathsf{NN}(x) > C\mathsf{NN}(c)$ since there is a path intersecting a finite number of tetrahedra between them. Since B is empty, $\mathsf{NN}(c) \geq R$ and hence $\mathsf{NN}(x) > CR$.

□

3.3 Radius-edge ratio

The radius-edge ratio is defined as follows:

Definition 3.3.1 (Radius-edge ratio ρ) *The radius-edge ratio of a tetrahedron is the ratio between the radius R of its circumscribed sphere and the length of its shortest edge e, i.e.*

$$\rho = \frac{R}{e}.$$

To understand the necessity of introducing the radius-edge ratio, consider Figures 3.2 and 3.3 that classify the bad aspect ratio tetrahedra into two categories. Let E be the largest edge of a tetrahedron, e the smallest edge of the tetrahedron and R its circumradius. The bad aspect ratio tetrahedra can be classified using the radius-edge ratio ρ, and the largest to smallest edge ratio $\mathcal{E} = \frac{E}{e}$.

The tetrahedra of Figure 3.2 have large $\mathcal{E}$. Note that a large $\mathcal{E}$ implies a large ρ since $2R \geq E$. Tetrahedra with large $\mathcal{E}$ ratio can not belong to a well-spaced point set, since the two nodes incident to the shortest edge are spaced very closely compared to the two nodes incident to the longest edge. This analysis only applies when the circumscribed sphere is empty, as is true in the case of a Delaunay triangulation. The tetrahedra of Figure 3.3 have $\mathcal{E} = O(1)$. If ρ is very large, they can not be Delaunay tetrahedra of a well-spaced node set, because the empty Delaunay circumsphere forms a large gap. The only bad aspect ratio tetrahedron that is truly well-spaced with respect to a Delaunay triangulation is the sliver of Figure 3.3(c). This sliver has a bounded radius-edge ratio.

Such slivers can appear in the Delaunay diagram of well-spaced points. Even if the point set is very regular, such as the nodes of a cubical grid, numerical error can introduce slivers on the faces of the grid's sub-cubes. Recently, Chew [22] presented an algorithm that generates good aspect ratio Delaunay meshes. The theoretical bounds on the aspect ratio of tetrahedra generated by the algorithm are fairly large, though that does not necessarily imply the aspect ratios observed in practice will be large. Currently, however, no experimental data is publicly available. Furthermore, later computations over a sliver-free mesh might introduce slivers because of numerical error, as pointed above. Slivers are hard to avoid, and the approach taken here is to tame slivers rather to go to the expense of removing them.

The wish for the ability to handle slivers is the geometrical motivation for investigating Delaunay meshes with slivers, i.e. Delaunay meshes with bounded radius-edge ratio. A numerical motivation to defining the radius-edge ratio and investigating bounded radius-edge ratio Delaunay meshes is

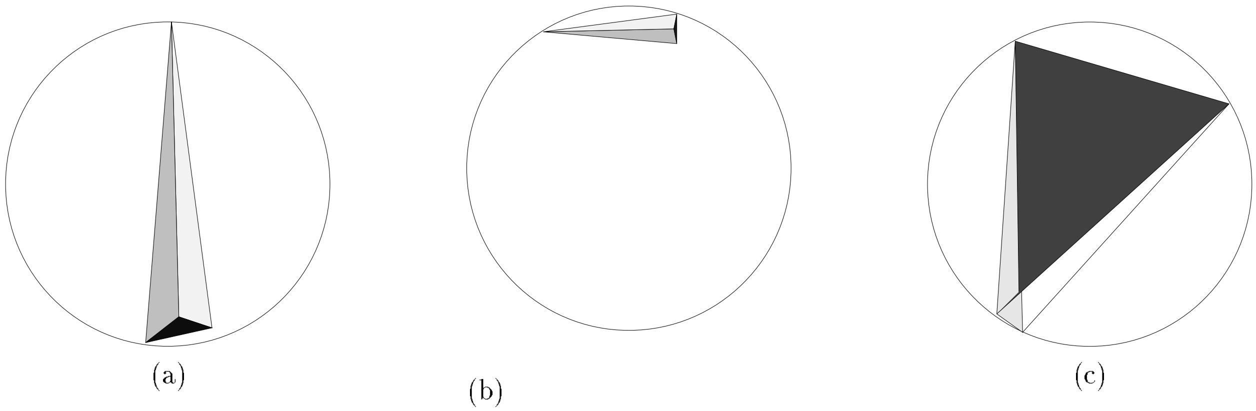


Figure 3.2: Let E be the largest edge of a tetrahedron, e its smallest edge and R its circumradius. The tetrahedra in (a),(b) and (c) are examples for bad aspect ratio tetrahedra with a large $\mathcal{E} = \frac{E}{e}$ ratio. Note that $\rho = \frac{R}{e}$ must be large as well in this case.

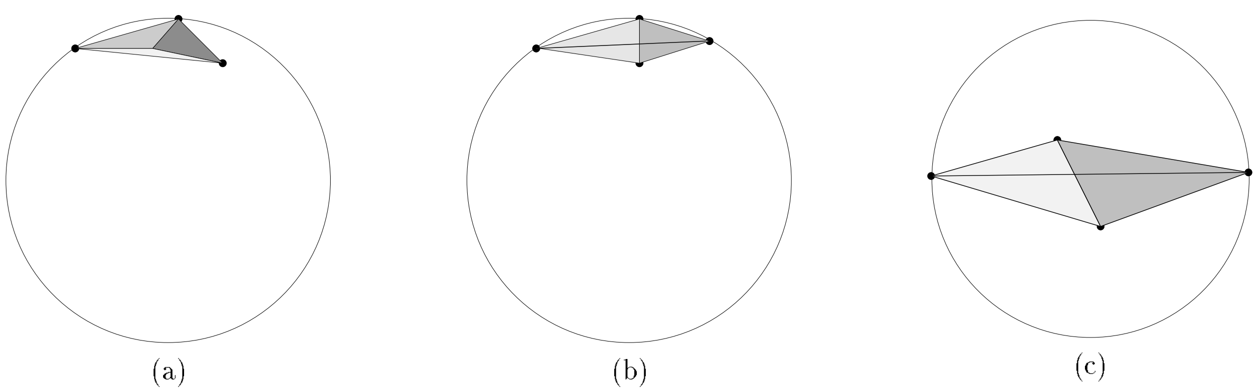


Figure 3.3: Examples for bad aspect ratio tetrahedra with $\mathcal{E} = O(1)$. Tetrahedra (a) and (b) also have large ρ, but (c) is of $\rho = O(1)$. A tetrahedron with both $\mathcal{E} = O(1)$ and $\rho = O(1)$ is called a **sliver**. It is the only bad aspect ratio tetrahedron with a bounded radius-edge ratio. A sliver is characterized by four almost planar points, nice triangular faces and a small circumscribed sphere. Note that slight perturbations of the nodes of the sliver can transform it to a tetrahedron like (b) with large ρ. The tetrahedron (a) is called a **cap** and is composed of a node centered above a triangular base.

the convergence behavior of the control volume meshes over such meshes. As Theorem 1.2.2 of Section 1.2.3 states, a bounded radius-edge ratio is sufficient to get optimal rates of convergence for approximate solutions of Poisson's equation constructed using control volume techniques.

3.4 Density graphs

Let $M = (\mathcal{P}, E, \mathcal{B})$ be a Delaunay mesh. Let ρ be a constant bound on the radius-edge ratio of M. Showing that the 1D skeleton of M is a density graph is the main important step towards showing its nodes are well-spaced.

Definition 3.4.1 (Density graph) *Let $M = (\mathcal{P}, E, \mathcal{B})$ be a mesh. The 1D skeleton of M is a $\mathcal{C}$-density graph if for every node $p \in \mathcal{P}$,*

$$\frac{||p-u||}{||p-v||} \leq \mathcal{C}$$

where u is the mesh node closest to p, and v is the farthest node from p that shares a mesh edge with it.

For bounded aspect ratio meshes, it is easy to show their 1D skeleton is a density graph by using a volume argument: the solid angle of each tetrahedron incident to p is larger than some constant θ, hence there can be at most $\frac{2\Pi}{\theta}$ tetrahedra incident to p. Since the edge lengths of neighboring tetrahedra can vary by at most a constant factor $C(\theta)$, the ratio $\frac{E}{e} < C(\theta)^{\frac{2\Pi}{\theta}}$. The tetrahedra are of bounded aspect ratio, therefore the smallest edge at a node is on the same order as the distance to the nearest node. Let ℓ be the distance to the node closest to p, then $\ell > ae$ and $\frac{E}{\ell} < \frac{C(\theta)^{\frac{2\Pi}{\theta}}}{a}$

A mesh with slivers can contain a tetrahedron with a very small solid angle and therefore a volume argument can not be used. The argument to show the Delaunay 1D-skeleton is a density graph proceeds using the Delaunay properties of the tetrahedra.

Lemma 3.4.2 *Let $M = (\mathcal{P}, E, \mathcal{B})$ be a mesh with radius-edge ratio bounded by ρ. Let $p \in \mathcal{P}$, and let T_1 and T_2 be two tetrahedra incident to p, their respective radii r_1 and r_2, and their centers c_1 and c_2. Assume the angle between the two radials, (p, c_1) and (p, c_2), is smaller than $\theta = \frac{1}{2}\arcsin(\frac{1}{2\rho})$. Then for the constant $\mathcal{C}_1 = \frac{1}{\sin\theta}$, dependent on ρ only,*

$$\frac{r_1}{r_2} \leq \mathcal{C}_1$$

Proof: Let B_i be the Delaunay circumsphere of T_i. Without loss of generality, assume $r_1 > r_2$. Look at Figure 3.4. Let $\alpha < \theta$ be the angle between the two radials. Let q be the point diametrically opposite to p on the circle $B_1 \cap B_2$. Let L be the length of the segment (p, q). Looking at the triangles $(p, c_1, 0.5 * (p+q))$ and $(p, c_2, 0.5 * (p+q))$, clearly $L/2 = r_1 \sin(\beta) = r_2 \sin(\alpha+\beta)$, so

$$\frac{r_1}{r_2} = \frac{\sin(\alpha+\beta)}{\sin\beta} \leq \frac{1}{\sin\beta}$$

The nodes of T_2 must be outside B_1, hence at least one of T_2's edges is smaller than $L = 2r_2\sin(\beta+\alpha)$. The bounded radius-edge ratio now implies $r_2/(2r_2\sin(\beta+\alpha)) \leq \rho$. Hence $\sin(\beta+\alpha) \geq \frac{1}{2\rho}$ and $\beta+\alpha \geq \arcsin\frac{1}{2\rho}$. Since $\alpha < \frac{1}{2}\arcsin\frac{1}{2\rho}$, this means $\beta \geq \frac{1}{2}\arcsin\frac{1}{2\rho} = \theta$. Therefore,

$$\frac{r_1}{r_2} \leq \frac{1}{\sin\beta} \leq \frac{1}{\sin\theta}$$

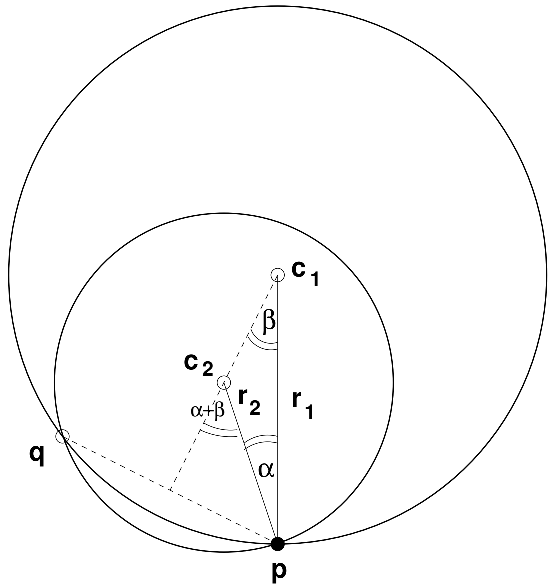


Figure 3.4: A 2D view of two Delaunay spheres incident to p, on the plane defined by their radials from p.

□

Lemma 3.4.3 *Let T_1, T_2 be two tetrahedra of a Delaunay mesh with a radius-edge ratio bounded by ρ. Let E_i be the largest edge of tetrahedron T_i, e_i the smallest edge and r_i the radius of the circumscribing sphere.*

1. $\frac{E_i}{e_i} \leq 2\rho$
2. *If T1 and T2 are neighbors then* $\frac{E_1}{e_2} \leq 4\rho^2$.
3. *If* $\frac{r_1}{r_2} \leq C$ *then* $\frac{E_1}{e_2} \leq 2C\rho$.

Proof:

1. The largest edge $E_i \leq 2r_i$ hence $\frac{E_i}{e_i} \leq \frac{2r_i}{e_i} = 2\rho$.
2. Let e be an edge common to T_1 and T_2. $\frac{E_1}{e_2} \leq \frac{E_1}{e}\frac{e}{e_2} \leq 4\rho^2$.
3. $\frac{E_1}{e_2} \leq \frac{E_1}{r_1}\frac{r_1}{r_2}\frac{r_2}{e_2} \leq 2C\rho$

□

The following is the main theorem, showing that the 1D skeleton of a bounded radius-edge ratio mesh is a density graph. Note that it suffices to show that the ratio E/e of the largest edge to the smallest edge is bounded, since the Delaunay diagram always contains the edges between any node and its closest node.

Theorem 3.4.4 *Let $M = (\mathcal{P}, E, \mathcal{B})$ be a Delaunay mesh. Let ρ be a constant bound on the radius-edge ratio of M. There exists a constant C, depending on ρ only, such that* $\frac{E}{e} < C$.

Proof: To show the Delaunay triangulation is a density graph, cover a very small sphere S centered at a point $p \in P$ by a collection of circular patches with cone angle $\theta = \frac{1}{2}\arcsin(\frac{1}{2\rho})$, the angle defined in Lemma 3.4.2. There is a cover of S with no more than a constant C_3 such circular patches.

Let $C_2 = \max(2C_1\rho, 4\rho^2)$. C_1 is the constant of Lemma 3.4.2, and C_2 is fixed to accommodate the cases in Lemma 3.4.3. Each radius vector from p intersects sphere S in at least one cone patch (the patches are not necessarily disjoint, so it could intersect more than one patch). Assign to each radius vector a label which corresponds to one of the patches it intersects. If two radius vectors have the same label, then by Lemmas 3.4.3 and 3.4.2, the maximal ratio of the edges belonging to the two tetrahedra is bounded by C_2.

Let e and E be the shortest and the longest Delaunay edges, respectively, incident to p. There is a path between e and E through neighboring tetrahedra incident to p. In each transition of the path, the edge lengths can grow by at most a factor of C_2.

Assign a label to each tetrahedron in the path. The label indicates the patch that the tetrahedron's radius vector intersects. If a label appears more than once in the path, the path can be shortcut by removing all labels between last and first appearance of the label, and instead using the ratio information forced by the label, which is C_2. This reduces the number of labels to less then C_3 because no label can repeat. Therefore the ratio of E to e is bounded by $C = C_2^{C_3}$ and hence $\mathcal{P}$ is a C-density embedding of $DT(\mathcal{P})$. □

Lemma 3.4.5 *The vertex degree of each node in an 3D C-density graph (i.e. $E/e < C$ when E and e are incident on the same node) is bounded by* $(2C^2+1)^3$

Proof: Let $p \in \mathcal{P}$. A neighboring node q of p is contained in the sphere with radius $R = Ce$ centered at p, where e is the smallest edge incident to p. q has an edge of length at least $|e|$, so q's nearest neighbor is no closer than $|e|/C$. Therefore, the sphere centered at q of radius $r = |e|/(2C)$ does not intersect with the sphere centered at any other neighboring node of p of radius r. These smaller sphere are contained in a sphere of radius $R + r$ centered at p. By a volume argument, there can be no more than

$$\frac{(R+r)^3}{r^3} = (2C^2+1)^3$$

such spheres. □

3.5 The shape of the Voronoi diagram

The previous section showed that the skeleton of a bounded radius-edge ratio Delaunay diagram is a density graph. This has interesting implications for the shape of the finite Voronoi cells in the dual Voronoi diagram. In particular, they are well-shaped in a manner reminiscent of a standard bounded aspect ratio triangulation: the ratio between the radius of the smallest containing sphere and the radius of the largest contained sphere is bounded. Informally, this means the cells resemble spheres, and are not long and skinny. The slivers in the Delaunay triangulation can cause long and skinny faces to appear on the face of the Voronoi cell but do not affect its overall shape. It is this property that makes a Delaunay diagram with slivers so similar to a bounded aspect ratio mesh.

Theorem 3.4.4 and Lemma 3.4.5 are now used to show that the Voronoi diagram is well-shaped.

Lemma 3.5.1 *Let $M = (\mathcal{P}, E, \mathcal{B})$ be a Delaunay mesh. Let ρ be a constant bound on the radius-edge ratio of M.*

Let $\mathsf{NN}_{\mathcal{P}}$ be the nearest neighbor function. (Definition 2.2.4). Let $V(p)$ be a finite Voronoi cell of the Delaunay diagram, $p \in \mathcal{P}$ the mesh node associated with the cell.

Then:

1. $\mathsf{B}(p, \frac{1}{2}\mathsf{NN}(p))$, *the sphere centered at p with radius $\frac{1}{2}\mathsf{NN}(p)$, is contained in V.*
2. $\mathsf{B}(p, \rho C\mathsf{NN}(p))$, *the sphere centered at p with radius $\rho C\mathsf{NN}(p)$, contains V.*

Proof:

1. The distance to the nearest neighbor of p is also the length of the shortest edge e incident to p. The Voronoi diagram is the intersection of all the bisector planes between p and its neighboring nodes. None of these planes is closer than $NN(p)/2$ to p, hence it must contain the sphere $\mathsf{B}(p, \mathsf{NN}(p)/2)$.
2. The distance to the nearest neighbor of p is also the length of the shortest edge e incident to p. By Theorem 3.4.4 the largest edge incident to p is E such that $E < Ce$. C is a constant depending on ρ only. Furthermore, by the radius-edge ratio, the largest radius of a Delaunay circumsphere at p is smaller than $\rho Ce = \rho C\mathsf{NN}(p)$. Clearly the sphere $\mathsf{B}(p, \rho C\mathsf{NN}(p))$ that contains all the corners of the Voronoi cell, contains the cell itself.

□

Infinite Voronoi cells can occur on the convex hull of the point set, or the boundary. Clearly, the statement of Lemma 3.5.1 is valid for a truncated form of the infinite Voronoi diagram, that includes only the intersection of Voronoi cell around p with the outer ball $\mathsf{B}(p, \rho C\mathsf{NN}(p))$. This truncated form of the cell contains all the corners of the Voronoi cell.

The following lemma discusses the values the nearest neighbor function can take over the Voronoi cells and the edges of the Delaunay diagram.

Lemma 3.5.2 *Let $M = (\mathcal{P}, E, \mathcal{B})$ be a Delaunay mesh with a constant bound ρ on the radius-edge ratio of M. Let $\mathsf{NN}_{\mathcal{P}}$ be the nearest neighbor function of $\mathcal{P}$.*

Let $V(p)$ be a finite Voronoi cell of the Delaunay diagram, $p \in \mathcal{P}$ the Mesh node associated with the cell. Let $x \in V(p)$ be an arbitrary point in the Voronoi cell. Then,

$$\frac{1}{4}\mathsf{NN}(p) \leq \mathsf{NN}(x) \leq (1 + \rho C)\mathsf{NN}(p)$$

Proof: $x \in V(p)$ implies that x is closer to p than to any other node of $\mathcal{P}$. The value $\mathsf{NN}(x)$ is the radius of the smallest sphere containing two points of $\mathcal{P}$. One of them must be p, let the other be q. Clearly, $\mathsf{NN}(x) \leq \|x - p\| + \mathsf{NN}(p)$. By Lemma 3.5.1, $\|x - p\| \leq \rho C\mathsf{NN}(p)$, hence $\mathsf{NN}(x) \leq (1 + \rho C)\mathsf{NN}(p)$. For the other inequality, look at the distance $\|x - p\|$:

- if $\|x - p\| > \mathsf{NN}(p)/4$, then $\mathsf{NN}(x) \geq \mathsf{NN}(p)/4$.
- if $\|x - p\| \leq \mathsf{NN}(p)/4$, then look at the sphere $\mathsf{B}(p, \mathsf{NN}(p)/2)$. By Lemma 3.5.1, this sphere is fully contained in the Voronoi cell, hence point q is outside this sphere. This implies $\|x - q\| \geq \mathsf{NN}(p)/4$, and therefore $\mathsf{NN}(x) \geq \mathsf{NN}(p)/4$.

□

Note that Lemma 3.5.2 is valid also over the truncated Voronoi cell.

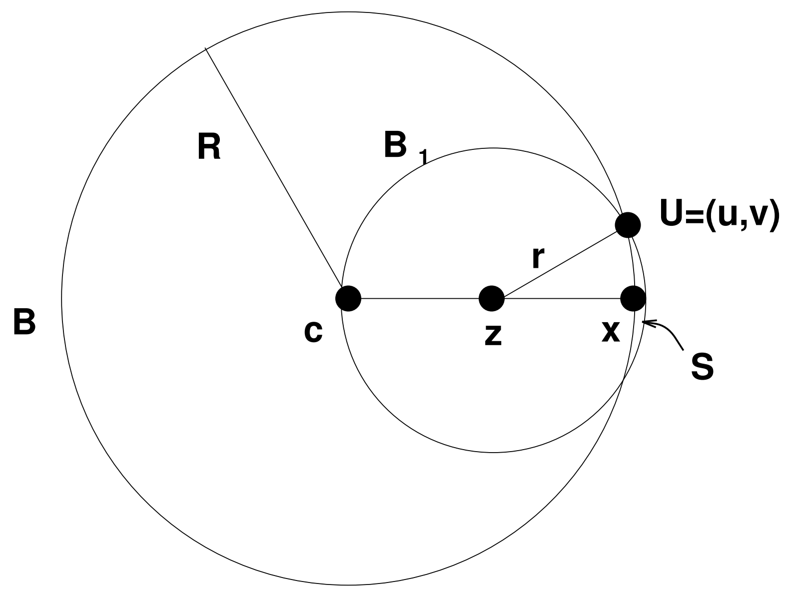


Figure 3.5: Illustration for Lemma 3.6.1.

3.6 Bounded radius-edge ratio meshes have well-spaced nodes

This section finally shows that the gaps in a bounded radius-edge ratio Delaunay mesh are bounded with respect to the function $\mathsf{NN}_{\mathcal{P}}$, and thus that its nodes are well-spaced with respect to $\mathsf{NN}_{\mathcal{P}}$.

Lemma 3.6.1 *Let $M = (\mathcal{P}, E, \mathcal{B})$ be a bounded radius-edge ratio tetrahedral mesh. Let ρ be a bound on the radius-edge ratio. Let B be a an interior gap with center c. Let x be a point on the circumference of B such that the radial $(x, c) \in \Omega$.*

If B does not contain any of the nodes $\mathcal{P}$, then the number of Voronoi cells intersecting the radial (x, c) is bounded by a constant depending only on ρ.

Proof: Let R be the radius of the sphere B. Let z be a point on the radial at distance $R/2$ from c. z belongs to the Voronoi cell of some point p. Also, since $\|p - z\| \geq R/2$, Lemma 3.5.1 implies that

$$\rho C \mathsf{NN}(p) \geq \|p - z\| \geq R/2$$

hence $\mathsf{NN}(p) \geq R/(2\rho C)$, and the Voronoi cell of p contains the sphere $\mathsf{B}(p, R/(4\rho C))$.

Look at Figure 3.5. Assume the radial is on the x axis, and that c is the origin. Consider a sphere B_1 of radius $R/2 + \epsilon$ centered at z. The 2D projection of B_1 and B intersect at two points. Let $U = (u, v)$ such that $v > 0$ be one of the points. Solving for U, it is easy to see that

$$u = R - \epsilon - \epsilon^2/R \qquad v \leq \sqrt{2R\epsilon}$$

Therefore, by setting $\epsilon = \frac{R}{128\rho^2 C^2}$, it holds that $v \leq \frac{R}{8\rho C}$.

Consider where can the point p be located. Let $S = B_1 \setminus B$. If $p \in S$, then using the fact that the ball $B_2 = \mathsf{B}(p, R/(4\rho C))$ is contained in the Voronoi diagram of p, combined with the fact that $v \leq \frac{R}{8\rho C}$, it can be concluded that $S \subset B_2$. In that case no point $q \in \mathcal{P}$ can be in S. On the other hand, if $p \notin S$, again no point $q \in \mathcal{P}$ can be in S, since then q would be closer to z than p is, a contradiction to the definition of the Voronoi cell.

To conclude, S is either empty or contains a point p with a large Voronoi cell relative to R. Therefore, no point q with a small Voronoi cell can get close to the radial. The radial intersects only Voronoi cells whose center point q has $\mathsf{NN}(q) > cR$ for some positive constant c. Therefore, at most a constant number of truncated Voronoi cells can intersect the radial.

□

Theorem 3.6.2 *Let $M = (\mathcal{P}, E, \mathcal{B})$ be a bounded radius-edge Delaunay mesh, Let $\mathsf{NN}_{\mathcal{P}}$ be its nearest neighbor function. There exist constants $0 < \mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$ such that with respect to the function $\mathsf{NN}_{\mathcal{P}}$:*

1. *edge boundary gaps are bounded by $\mathcal{L}_1$*
2. *face boundary gaps are bounded by $\mathcal{L}_2$*
3. *interior gaps are bounded by $\mathcal{L}_3$*

i.e. $\mathcal{P}$ is 2-spaced according to $\mathsf{NN}_{\mathcal{P}}$ and has the bounded gaps property with parameter

$$\mathcal{L} = \max(\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3).$$

Proof:

1. Similar to Theorem 2.4.5, using the fact the graph is a density graph.
2. Let F be a boundary face. M restriceted to F is a good aspect triangulation on F, hence the 2D corresponding theorem 2.4.4 can be used to derive the result.
3. Using Lemma 3.6.1. Let $x \in \Omega$ be an arbitrary point, B an interior gap through x. Let c be the center of B, R its radius. By Lemma 3.6.1, $\mathsf{NN}(x) > C\mathsf{NN}(c)$ since there is a path intersecting a finite number of Voronoi cells between them. Since B is empty, $\mathsf{NN}(c) \geq R$ and hence $\mathsf{NN}(x) > CR$.

□

3.7 Summary

This chapter generalized to three dimensions the first part of the results of chapter 2: a good aspect ratio mesh has well-spaced point set. Furthermore, the radius-edge ratio was introduced, and the following properties of bounded radius-ratio Delaunay meshes have been shown:

- their 1D skeleton is a density graph,
- their Voronoi cells are well-shaped, and
- their nodes are well-spaced points.

Delaunay meshes with bounded radius-edge ratio are important in the context of the control volume method, and because the Delaunay triangulation of well-spaced points leads to a mesh with bounded radius-edge ratio, but not a bounded aspect ratio since thin and flat elements called slivers may appear. In some sense, the equivalent of well-spaced points in three dimensions is a bounded radius-edge ratio mesh.

Chapter 4

Generating 3D Meshes From Well-Spaced Points

This chapter presents both an algorithm and an algorithmic framework for generating a bounded radius-edge Delaunay mesh that is conforming to a given non-manifold three dimensional domain and that is of optimal size in a sense to be defined below.

Chapter 2 showed that a constrained Delaunay triangulation of a well-spaced point set in two dimensions is of bounded aspect ratio if set points are not too close to a boundary segment they are not incident to.

This result can not be generalized to three dimensions because a Delaunay tetrahedralization of well-spaced points is not necessarily of bounded aspect ratio since slivers may arise, see Section 3.3. Instead, the goal here is to generalize the result to bounded *radius-edge* ratio Delaunay meshes.

However, an attempt to obtain this generalization must overcome the the following two obstacles:

- There is no reasonable definition of a constrained Delaunay triangulation in 3D. Furthermore, there are examples of domains that can not be tetrahedralized, see Figure 4.1.

- Unless extra measures are taken, bad radius-edge ratio tetrahedra can appear on the boundary of well-spaced points, see Figure 4.2. A similar problem in two dimensions is solved by adding the restriction that points may not be placed too close to boundary segments they are not incident to. In 3D this additional restriction no longer suffices and additional restrictions are necessary.

This chapter shows that the above problems can be circumvented. Given a domain, the algorithm presented here constructs well-spaced points whose Delaunay triangulation conforms to the domain, is of bounded radius-edge ratio, and has the further property that the Delaunay centers of circumspheres are always guaranteed to reside inside the domain. Since the resulting triangulation is conforming, the first obstacle is avoided, and since Delaunay centers are within the domain, the kind of tetrahedron mentioned in the second obstacle can not arise.

Section 4.1 surveys related work in mesh generation and tetrahedralization of domains and point sets in three dimensions. Section 4.2 defines the boundary descriptions allowed as input. Section 4.3 presents the algorithm and a proof for its correctness, in particular showing that the mesh is a bounded radius-edge ratio mesh conforming to the domain description, whereas Section 4.4 bounds the number of elements in the generated mesh.

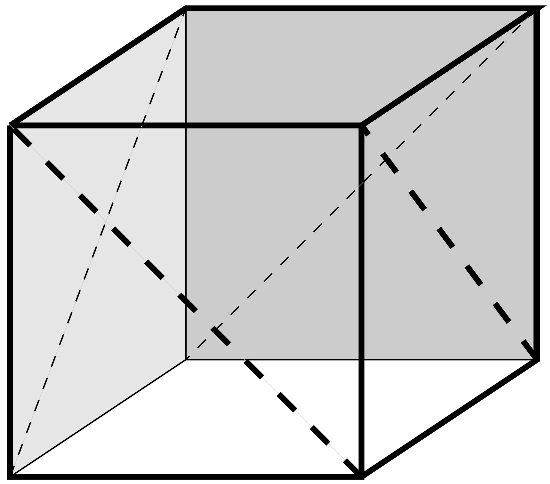


Figure 4.1: A domain that can not be tetrahedralized. This example is based on an example due to Schönhardt [66]. Starting from a cube, twist the top square clockwise very slightly. As a result, the dashed lines appear as edges with a reflex angle. This twisted cube can not be tetrahedralized because the standard cube has no decomposition into tetrahedra that contains all the dashed edges.

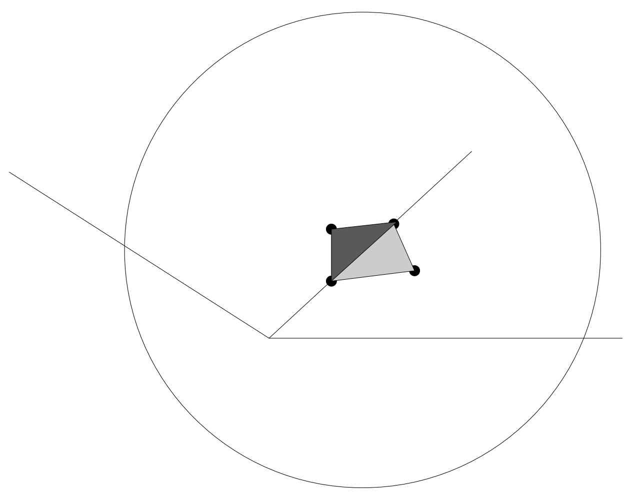


Figure 4.2: Bad radius-edge ratio tetrahedra can occur on the boundary of the triangulation of well-spaced unless special measures are taken, such as not allowing points on the faces to be inside the dihedral sphere of the boundary segments' edges. The figure shows two boundary faces meeting at a large, almost flat, dihedral angle. The circumscribed sphere of a tetrahedron formed as drawn here can be arbitrarily large. However, the interior of this large sphere is mostly below the two planes and thus outside of the domain. Only a thin layer on the top of the sphere is in the domain.

4.1 Background and previous results

In this section the properties of three dimensional point sets and polyhedral domains and their partitionings into tetrahedra are discussed. The wonderful survey on mesh generation and triangulations by Bern and Eppstein [8] has a section on three dimensions that covers in more details many of the results mentioned here.

Tetrahedralizing point sets in 3D: Three dimensional tetrahedralizations are surprisingly more complex than their counterparts in two dimensions. Given a point set with n points, a tetrahedralization of its convex hull may have anywhere from $O(n)$ to $O(n^2)$ tetrahedra. This is true even for the Delaunay diagram of the points: there exist some point sets whose Delaunay triangulation has $O(n^2)$ elements. Nonetheless, any point set can be tetrahedralized using only $O(n)$ tetrahedra. This is done by first tetrahedralizing the convex hull using $O(n)$ tetrahedra, and then adding the internal points one by one. As a point is added, the tetrahedron that contains it is split into four tetrahedra.

The Delaunay tetrahedralization in 3D is unfortunately not as well suited as it is in 2D. Most importantly, it does not maximize the smallest (dihedral or solid) angle in the tetrahedralization. However, Rajan [60] showed the Delaunay triangulation minimizes the maximum radius of a min-containment sphere. The min-containment sphere of an element is the smallest sphere containing the element, and is not necessarily equal to the circumscribed sphere. For the goal of generating good aspect ratio meshes the min-containment property is not as useful as the two dimensional Delaunay optimality property. Nonetheless, the min-containment sphere is related to the radius-edge ratio, especially when the containment sphere is contained inside the domain, which implies the relevance of the Delaunay triangulation for bounded radius-edge ratio mesh generation.

The 3D equivalence of two dimensional edge flipping is not guaranteed to converge to the Delaunay diagram if one starts from an arbitrary tetrahedralization. However, Joe [37] gives a flipping-based algorithm to incrementally construct the Delaunay diagram. He shows that if a point is added to an existing Delaunay diagram, flipping does converge to the new Delaunay diagram of the revised point set. Rajan [60] also supplies an incremental flipping-based approach.

The Delaunay tetrahedralization can be computed directly, using four dimensional convex hull algorithms, or incrementally using Watson's algorithm [73]. Chan, Snoeyink and Yap [16] gave an output-sensitive algorithm for Delaunay diagrams computations. Since the size of the Delaunay diagram varies from $O(n)$ to $O(n^2)$, the complexity of their algorithm can be much smaller than the worst case optimal quadratic complexity of other algorithms.

Tetrahedralizing domains in 3D: Not all polyhedral domains are tetrahedralizable. An example for such a domain was given in Figure 4.1. In fact, Ruppert and Seidel [65] showed that it is NP-complete to decide if a polyhedron is tetrahedralizable, or how many Steiner points are necessary to tetrahedralize it. Some domains may require $O(n^2)$ Steiner points to be added before they can be tetrahedralized [18]. The closer the polyhedron is to a convex polyhedron, the fewer Steiner points are necessary. The number of reflex angles in a polyhedron is a measure of how close to being convex the polyhedron is. Chazelle and Palios [19] have shown that any simple polyhedron with n nodes and r reflex edges can be partitioned into $O(n + r^2)$ elements using $O(n + r^2)$ Steiner points.

Provably good mesh generation: The problem of provably good mesh generation is that of generating a mesh conforming to a given (polyhedral or piecewise linear) domain in 3D with certain optimality guarantees for the size and shape of the mesh elements. In particular, the element shape quality should be better than some given or proven lower bound, and the number of elements should be as small as possible up to a constant factor. For the finite element method, the element quality measure is its aspect ratio. For the control volume method, the quality measure is the radius-edge ratio (see Section 1.2).

Mitchell and Vavasis [52] gave a provably good mesh generation algorithm that produces bounded aspect ratio meshes. Their approach is a generalization of the two dimensional quadtree approach of Bern, Eppstein and Gilbert [9]. For a long time this approach was the only one proven to generate bounded aspect ratio meshes in 3D. The two dimensional alternative to quadtree mesh generation, suggested by Chew [21] and Ruppert [64], is based on the Delaunay triangulation and did not seem to generalize to three dimension because of the problem of slivers (see Section 3.3). Recently, however, Chew [22] gave a method to remove slivers from a Delaunay based three dimensional mesh.

The problem of generating provably good radius-edge ratio Delaunay meshes is interesting not only for the control volume method but also as a preparatory step before applying Chew's sliver removal step. Miller, Talmor, Teng, Walkington and Wang [50] gave a framework for generating such meshes, which is presented in some more detail in this chapter and is based on the idea of sphere packing. Shewchuk [67] generalized Ruppert's incremental approach to three dimensions to solve the same problem. Both methods produce a bounded radius-edge ratio mesh whose number of elements is at most a constant factor away from the number of elements in an optimal bounded aspect ratio mesh. This is reasonable if the intention is to merely tolerate slivers rather than use slivers whenever possible to reduce the mesh size. (See the discussion in Section 4.4 below).

4.2 The boundary description

A *polytope* is the convex combination of finite set of points P. The dimension of the polytope is the dimension of the affine subspace generated by P. Polytopes of dimension i are called i-polytopes. The boundaries as well as the domain considered are a collection of polytopes, as in the following definition:

Definition 4.2.1 *A* ***piecewise linear system*** *is a set of polytopes $\mathcal{B}$ with the following properties:*

1. *The set $\mathcal{B}$ is closed under taking boundaries, i.e., for each $P \in \mathcal{B}$ the boundary of P is a union of polytopes in $\mathcal{B}$.*

2. *$\mathcal{B}$ is closed under intersection.*

3. *Let $P, Q \in \mathcal{B}$. If $dim(P \cap Q) = dim(P)$ then $P \subseteq Q$, and $dim(P) < dim(Q)$.*

The domain is the 3-polytopes of the piecewise linear system, its boundaries the lower dimensional polytopes. The boundary description is further restricted by requiring 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 [2], but is more general. For example, a box with a segment in the interior can be described as a piecewise linear system with no need to further decompose it.

4.3 Algorithm "packing"

Definition 4.3.1 (point-ball) *The point-ball of a node p is a ball $\mathsf{B}(p, \frac{\mathsf{lfs}_{\mathcal{B}}(p)}{\beta})$.*

Algorithm: PACKING($\mathcal{B}, \beta$)

Input: $\mathcal{B}$, a piecewise linear system of dimension three
β, the spacing-constant used with lfs to set the radius of point-balls

Output: A set of points $\mathcal{P}$ and a tetrahedralization of $\mathcal{P}$ that conforms to $\mathcal{B}$

Method:

1. Let P_0 be of the 0-polytopes of $\mathcal{B}$
   Let $B_0 = \emptyset$

2. **For** $i = 1$ to 3

   For each i-polytope $D \in \mathcal{B}$

   i. For $j < i$: Let $P_j^D = P_j \cap D$ and $B_j^D = \{B(c, R) \in B_j \quad | \quad c \in D\}$.

   ii. Find a maximal set of points P_i^D in the interior of D such that

      A. $\cup_{k=0}^{i} P_k^D$ is β-spaced by $\text{lfs}_{\mathcal{B}}$

      B. P_i^D is disjoint from the circumscribing spheres in $\cup_{k=0}^{k=i-1} B_k^D$ and their interiors.

      C. If $i = 2$, $p \in P_i^D$ must be stay far from the 1-polytopes contained in D (See Definition 4.3.3).

   iii. Compute the Delaunay triangulation on D using only the points of $P_0^D \cup \cdots \cup P_i^D$. Set B_i^D to be the collection of the circumscribed balls of the resulting Delaunay partition and T_i^D the Delaunay triangles.

   end for

3. Set $B_i = \cup_D B_i^D$
   $P_i = \cup_D P_i^D$

   end for

4. Return $\mathcal{P} = P_0 \cup \cdots \cup P_3$ and $\mathcal{T} = \cup_D T_3^D$. $\mathcal{T}$, the resulting tetrahedralization, is the collection of all the tetrahedra generated independently on each 3-polytope D.

Figure 4.3: 3-dimensional sphere-packing algorithm.

Definition 4.3.2 (Circumscribed sphere) *The circumscribed sphere of a set of points S such that $1 < |S| \leq 4$ in 3D is the sphere with the smallest radius passing through all the points of S. This definitions leads to the following spheres according to the size of S:*

1. *If $S = \{p, q\}$ the circumscribed sphere is centered half way between p and q with radius $0.5\|p - q\|$.*
2. *If $S = \{p, q, r\}$ the circumscribed sphere is centered on the plane spanned by S, and the radius is equal to the radius of the lower dimensional Delaunay circumsphere on that plane.*
3. *If $S = \{p, q, r, t\}$ this sphere is the Delaunay circumsphere of the tetrahedron S.*

Definition 4.3.3 (Staying far from edges) *Let Q be a 2-polytope of $\mathcal{B}$, and $E \subset Q$ a 1-polytope of $\mathcal{B}$. $p \in Q$ is called **far** from E if the sphere $\mathsf{B}(p, 2\mathsf{lfs}_{\mathcal{B}}(p))$ and its interior do not intersect E.*

Algorithm PACKING of Figure 4.3 constructs a point-set β-spaced by $\mathsf{lfs}_{\mathcal{B}}$. It proceeds by increasing dimension of the boundary elements. Points are packed on each boundary element such that (1) point-balls do not intersect, and (2) the interiors of circumscribed spheres of the triangulation of lower dimensional boundary elements remain empty. Furthermore, points on the face are restricted to stay away from the edges bounding the faces.

Once the points are placed on a polytope $D \in \mathcal{B}$, the polytope is Delaunay triangulated. If D is a segment, the Delaunay triangulation is a partition of D into edges. If D is a face of $\mathcal{B}$, then the partition is a Delaunay triangulation on D. Otherwise, it is the standard Delaunay tetrahedralization.

Remarks: (1) Note that the tetrahedralization is performed independently on each polytope. The resulting triangulation is conforming to the domain description $\mathcal{B}$, however it is not necessarily equal to a Delaunay triangulation of the final point set $\mathcal{P}$. For example, if the domain was composed of two disjoint convex polytopes placed close to each other, their Delaunay triangulations may interact. (2) The maximality requirement can be relaxed. In the proofs the only condition relied upon is that the centers of the Delaunay circumspheres can not be added. Therefore, the algorithm can be stated incrementally, the point set on each polytope increased until no Delaunay circumcenters can be added. This leads to an incremental algorithm in three dimensions similar to Ruppert's [64].

The algorithm constructs the set of points independently on each polytope. The following theorem shows the correctness of this approach, by showing that the nodes placed independently on the polytopes obey some global restrictions:

Theorem 4.3.4 (Correctness of independence) *Let p be a point of some 3-polytope $D \in \mathcal{B}$. Let $U \in \mathcal{B}$ be the lowest dimensional polytope containing p. Let $V \in \mathcal{B}$ be another polytope that is contained in D. Assume $\beta > 3 + 2\sqrt{2}$. The following are true:*

- *p's point-ball does not intersect the point-ball of any point $q \in P_i^V$ for $1 \leq i < 3$*
- *p does not intersect the interior of any sphere of B_i^V*

that is, even when U and V are packed independently they do not interfere with each other.

Proof: The theorem is true by the statement of the algorithm if $U \subseteq V$ or $V \subseteq U$, therefore assume that the dimension of the intersection of U and V is smaller than the smaller dimension of both. This also implies that the dimension of either U or V is at most two. To prove the theorem, the following are shown:

1. Assume U and V are disjoint polytopes. Let q be some node on V. As long as $\beta > 2$ the point-balls can not intersect, since $\ell = \|p - q\| > \max(\mathsf{lfs}(p), \mathsf{lfs}(q))$.

2. Assume U and V intersect. Lemma 4.3.14 shows that as long as $\beta > \frac{1}{\sqrt{2}-1}$ p's point-ball does not intersect the point-ball of node $q \in V$.

3. Assume V is a 1-polytope and that U and V are disjoint. Let $S = \mathsf{B}(c, R) \in B_1^V$ be some circumscribed sphere centered on V. The goal is to show p and the interior of S are disjoint. Let $\ell = \|c - p\|$. Since U and V are disjoint, both $\mathsf{lfs}(c)$ and $\mathsf{lfs}(p)$ are smaller than ℓ. By Lemma 4.3.9, $R \leq \frac{2}{\beta-1}\mathsf{lfs}(c)$. Hence, if $\beta > 3$ then $R < \mathsf{lfs}(c) \leq \ell$ and p and the interior of S can not intersect.

4. Assume V is a 1-polytope and that U and V intersect. Let $S = \mathsf{B}(c, R) \in B_1^V$ be some circumscribed sphere centered on V. If the intersection of U and V is a point, then by the 90° restriction on the angle between U and V, p and the interior of S must be disjoint. Otherwise, the intersection equals V and contradicts the assumptions.

5. Assume V is a 2-polytope and that U and V are disjoint. Let $S = \mathsf{B}(c, R) \in B_2^V$. The goal is to show p and the interior of S are disjoint. Let $\ell = \|c - p\|$. Since U and V are disjoint, both $\mathsf{lfs}(c)$ and $\mathsf{lfs}(p)$ are smaller than ℓ. By Lemma 4.3.12, $R \leq \frac{2\sqrt{2}}{\beta-3}\mathsf{lfs}(c)$. Hence, if $\beta > 3 + 2\sqrt{2}$ then $R < \mathsf{lfs}(c) \leq \ell$ and p and the interior of S can not intersect.

6. Assume V is a 2-polytope and that U and V intersect. Let $S = \mathsf{B}(c, R) \in B_2^V$. The intersection of U and V can be either a point or an edge. If the intersection is at a point, then by the 90° restriction between U and V, p and the interior of S are disjoint. If the intersection is an edge, and the interior of S intersects U, it must intersect U inside some protective circumsphere on the edge. (Recall that by Theorem 4.3.6 the center of S lies on V). In that case p and the interior of S are disjoint.

□

Theorem 4.3.5 (Conforming) *The output tetrahedralization of algorithm* PACKING *conforms to the input domain description $\mathcal{B}$, and is a legal tetrahedralization of $\mathcal{B}$.*

Proof: In each 3-polytope D, each segment and face of D is partitioned into edges and triangles respectively that are maintained within some empty circumscribed sphere within the algorithm, and hence must appear in the final Delaunay tetrahedralization of D. The tetrahedralization of two 3-polytopes D_1 and D_2 that share a face F or a segment E must agree, since the segment or face is triangulated initially, and the final partition of D_1 and D_2 both agree with the triangulations of F or E. □

Theorem 4.3.6 (Circumcenters are internal to the domain) *Let T be a tetrahedron belonging to a decomposition of a 3-polytope D. The circumcenter of T is contained in D.*

Proof: The circumcenter of T is a vertex of the Voronoi cells of all the four nodes of T. The theorem is shown by showing that all the vertices of Voronoi cells are internal. We focus on the Voronoi cell of $q \in D$, one of the vertices of T.

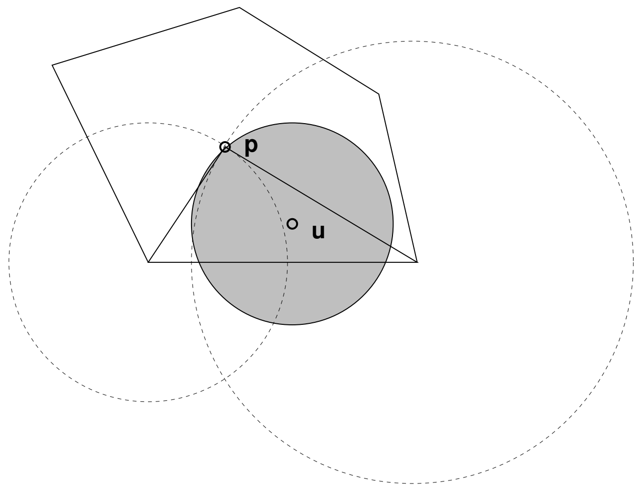


Figure 4.4: The dashed protecting spheres prevent any interior point from intruding on a plane Voronoi cell.

1. **When q is in the interior of D:** Let F, a 2-polytope of $\mathcal{B}$, be some face contained in D. We show that the Voronoi cell of q can not intersect F. Assume, by contradiction, that q's Voronoi cell does intersect F. Let u be a point in the intersection. After level two of the algorithm, u is in the Voronoi cell of some node $p \in F$. A contradiction is arrived at by showing that if u is closer to q than to p, then q is inside a circumscribed ball associated with a triangle of p. This is impossible by the algorithm construction. Divide the plane Voronoi cell of p into sectors. Concentrate on the sector that contains u (look at Figure 4.4). The claim is equivalent to showing that the gray shaded sphere around u is contained in the union of the dashed circles. This is true on the face F, and is extended to 3D by inspecting the figure as it would look on a plane vertical to F passing through any of the spheres.

2. **When q is on a face of D:** Let q reside on some 2-polytope F, a face of $\mathcal{B}$ that is contained in D. Following a similar reasoning to the previous case, the planar Voronoi cell of q on F does not intersect the edges of F, since otherwise q would encroach on some edge circumscribed sphere. Also, q's planar Voronoi cell can not intersect other faces of $\mathcal{B}$ without intersecting some edge.

   Similarly, when looking at q's 3D Voronoi cell, the following is true: it does not intersect any face that does not contain q (by the independence theorem q can not reside in any protective circumsphere, and hence the argument of the previous case applies), it does not intersect any edge, and it can contain only nodes from some 3-polytope D that q is on. Note that the algorithm tetrahedralized each 3-polytope separately, and that for each 3-polytope q is on it has a different Voronoi cell. q can not have Voronoi centers outside of D by convexity: assume by contradiction that q's D Voronoi cell has a center outside D. The center can not correspond to a tetrahedron of some internal node by the previous case. The center can not be joint to a node on another face by convexity. (The Voronoi cell of q outside of D must be vertical to F, and convexity means it can not meet a vertical to another face Voronoi cell). Nodes on the same face can not form a Voronoi center outside of D with q - hence the center can not exist.

3. **when q is on a segment of D:** Let q be a node on a segment E. The only case that remains is when all four nodes of the tetrahedron T are on segments of D, otherwise one of the previous cases apply. Let the other boundary edge be G. E and G can not be incident, nor can they belong to the same face F that contain E or the tetrahedron would be degenerate. Hence, by an argument similar to the first case, the Voronoi cell of p can not intersect F or any other face that q is on, and the circumcenter must be internal.

□

The following central theorem shows that the radius-edge ratio of the constructed tetrahedra is bounded:

Theorem 4.3.7 (Bounded radius-edge ratio) *Let $D \in \mathcal{B}$ be a 3-polytope. Let T be a tetrahedron in the Delaunay triangulation of D. The radius-edge ratio of a T is bounded by*

$$\frac{R}{d} \leq \rho = \frac{2\beta}{\beta - 7 - 2\sqrt{2}}$$

Proof: Let $S = \mathsf{B}(c, R)$ be the Delaunay circumsphere of T. Let ℓ be the shortest edge of T. By maximality, c can not be added to the point set. There can be several reasons c could not be added to the output node set in the course of the algorithm:

1. The point c is not in D. Theorem 4.3.6 shows this is impossible. Note that due to general position considerations, c can be assumed not to reside on a lower dimensional polytope.

2. The point-ball of c intersects some point-ball of a node in $p \in P \cap D$. Lemma 4.3.8 then implies that $\frac{R}{\ell} \leq \frac{\beta}{\beta-3}$.

3. c intersects the interior of some protective circumscribed sphere $B \in B_1^D$. Lemma 4.3.10 then implies $\frac{R}{\ell} \leq \frac{\beta\sqrt{2}}{\beta-3-2\sqrt{2}}$

4. c intersects the interior of some protective circumscribed sphere $B \in B_2^D$. Lemma 4.3.13 then implies $\frac{R}{\ell} \leq \frac{2\beta}{\beta-7-2\sqrt{2}}$

The last restriction gives the largest upper bound.

□

Lemma 4.3.8 *Let D be a 3-polytope of $\mathcal{B}$, and T one of its tetrahedra generated by algorithm* PACKING *with spacing parameter β. Let ℓ be the length of the shortest edge of T. Let $S = \mathsf{B}(c, R)$ be T's circumsphere. If the point-ball of the center c intersects the point-ball of some other point $b \in P \cap D$ in the packing:*

1. $\mathsf{lfs}(c) \geq R\frac{\beta-1}{2}$

2. $\frac{R}{\ell} \leq \frac{\beta}{\beta-3}$

Proof: See Figure 4.5. Let e be the shortest edge of T, and p and q the end-points of e. Let ℓ stand for the length of e. By the β-spacing, $\mathsf{lfs}(p) + \mathsf{lfs}(q) < \beta\ell$. Without loss of generality, $\mathsf{lfs}(p) < \beta\ell/2$. Denote by L the distance between b and c.

(1) b is on or outside c's sphere $\Rightarrow L \geq R$.

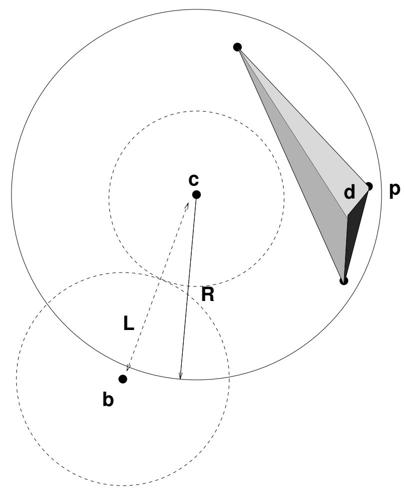


Figure 4.5: Lemma 4.3.8

(2) The spheres of b and c intersect $\Rightarrow \mathsf{lfs}(b) + \mathsf{lfs}(c) \geq \beta L$.

(3) lfs is Lipschitz $\Rightarrow \mathsf{lfs}(b) \leq \mathsf{lfs}(c) + L$.

(4) (2) and (3) $\Rightarrow 2\mathsf{lfs}(c) > L(\beta - 1) \geq R(\beta - 1)$.

(5) $\mathsf{lfs}(c) \leq \mathsf{lfs}(p) + R \Rightarrow R(\beta - 3)/2 \leq \mathsf{lfs}(p)$

(6) $\mathsf{lfs}(p) < \beta\ell/2$ and (5) $\Rightarrow \frac{R}{\ell} \leq \frac{\beta}{\beta-3}$

□

Lemma 4.3.9 *Let c be the center of a protective circumscribed sphere $S = \mathsf{B}(c, R) \in B_1^D$. Let q be some point inside S, then:*

1. $\mathsf{lfs}(c) \geq \frac{\beta-1}{2} R$
   Set $k_1 = \frac{\beta-1}{2}$
2. $\mathsf{lfs}(q) \geq \frac{\beta-3}{2} R = (k_1 - 1)R$

Proof: See the illustration in Figure 4.6.

1. Let a and b be the end points of the edge defining S, c is their midpoint. Assume, without loss of generality, that $\mathsf{lfs}(b) \geq \mathsf{lfs}(a)$.
   (1) maximality of packing on the edge, and the fact c can not be added to the edge $\Rightarrow$ $\mathsf{lfs}(b) + \mathsf{lfs}(c) > \beta R$
   (2) Lipschitz condition $\Rightarrow \mathsf{lfs}(b) \leq \mathsf{lfs}(c) + R$.
   (3) (1) and (2) $\Rightarrow 2\mathsf{lfs}(c) \geq R(\beta - 1)$
2. $\mathsf{lfs}(q) \geq \mathsf{lfs}(c) - R \geq R(\beta - 1)/2 - R$

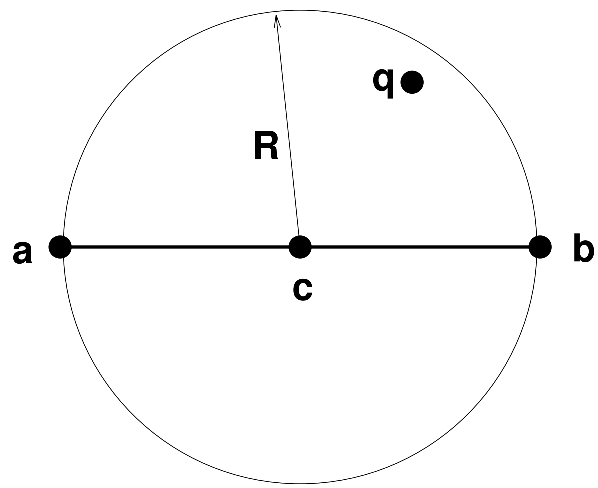


Figure 4.6: Lemma 4.3.9

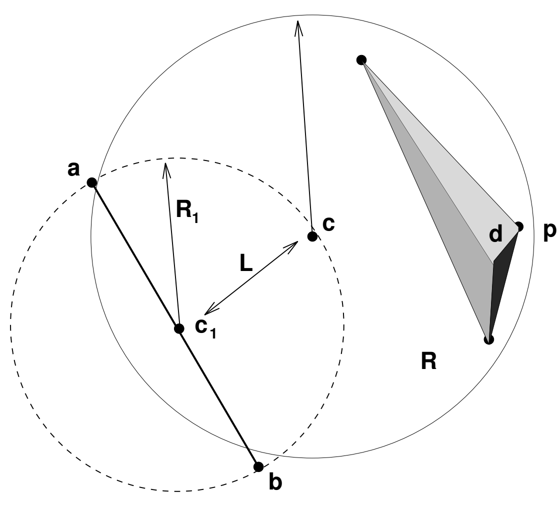


Figure 4.7: Lemma 4.3.10

□

Lemma 4.3.10 *Let D be a 3-polytope of $\mathcal{B}$, and T one of its tetrahedra generated by algorithm* PACKING *with spacing parameter β. Let ℓ be the length of the shortest edge of T. Let $S = \mathsf{B}(c, R)$ be T's circumsphere. If c intersects the interior of some protective circumscribed edge-sphere $S_1 = \mathsf{B}(c_1, R_1) \in B_1^D$ then:*

1. $\mathsf{lfs}(c) \geq R\frac{k_1-1}{\sqrt{2}} = R\frac{\beta-3}{2\sqrt{2}}$
2. $\frac{R}{\ell} \leq \frac{\beta\sqrt{2}}{2(k_1-1-\sqrt{2})} = \frac{\beta\sqrt{2}}{\beta-3-2\sqrt{2}}$

Where k_1 is fixed in Lemma 4.3.9

Proof: See Figure 4.7. Let e be the shortest edge of T, and p and q the end-points of e. Let ℓ stand for the length of e. By the β-spacing, $\mathsf{lfs}(p) + \mathsf{lfs}(q) < \beta\ell$. Without loss of generality, $\mathsf{lfs}(p) < \beta\ell/2$. Let L be the distance from c to c_1.

(1) c is in the protective-edge S_1, hence Lemma 4.3.9 $\Rightarrow \mathsf{lfs}(c) \geq (k_1 - 1)R_1$ and $L \leq R_1$

(2) Triangle inequality $\Rightarrow L^2 + R_1^2 \geq R^2 \Rightarrow R_1 \geq R/\sqrt{2}$.

(3) (1) and (2) $\Rightarrow \mathsf{lfs}(c) \geq R(k_1 - 1)/\sqrt{2}$.

(4) $\mathsf{lfs}(c) \leq \mathsf{lfs}(p) + R \Rightarrow \mathsf{lfs}(p) \geq \mathsf{lfs}(c) - R \geq R\frac{k_1-1-\sqrt{2}}{\sqrt{2}}$.

(5) $\mathsf{lfs}(p) < \beta\ell/2 \Rightarrow \beta\ell/2 \geq R\frac{k_1-1-\sqrt{2}}{\sqrt{2}}$ or

$$\frac{R}{\ell} \leq \frac{\beta\sqrt{2}}{2(k_1-1-\sqrt{2})} = \frac{\beta\sqrt{2}}{\beta-3-2\sqrt{2}}$$

□

Lemma 4.3.11 *Let $F \in \mathcal{B}$ be a 2-polytope. Let $S = \mathsf{B}(c, R)$ be a protective circumscribed sphere in B_2^F. If the only reason c was rejected from the packing is that it was too close to some 1-polytope $E \subseteq F$ (i.e. the ball $S_0 = \mathsf{B}(c, \frac{2\mathsf{lfs}(c)}{\beta})$ intersects E), then $\mathsf{lfs}(c) \geq \frac{\beta}{2\sqrt{2}}R$.*

Proof: Let q be some point on E such that q intersects S_0. Let S_1 be the protective circumscribed sphere of the sub-segment of E that q is on. By the assumption, c does not intersect the interior of $S_1 = \mathsf{B}(c_1, R_1)$. Let L be the distance between q and c. Assume, by way of contradiction, that $\frac{2\mathsf{lfs}(c)}{\beta} < \frac{R}{\sqrt{2}}$.

1. q is in $S_0 \Rightarrow L \leq \frac{2\mathsf{lfs}(c)}{\beta} < \frac{R}{\sqrt{2}}$
2. (1) implies $\Rightarrow q \in S$.
3. (1) and (2) $\Rightarrow$ the edge q is on intersects S, and the distance from c to the edge is smaller than L. Since the edge end points must be outside the empty ball S, this means $R_1 \geq R/\sqrt{2}$.
4. $R_1 \geq R/\sqrt{2}$ together with $L < \frac{R}{\sqrt{2}} \Rightarrow c$ is in S_1, a contradiction to the assumptions.

□

Lemma 4.3.12 *Let $F \in \mathcal{B}$ be a 2-polytope. Let c be the center of a protective circumscribed sphere $S = \mathsf{B}(c, R) \in B_2^F$. Let q be some point inside S, then:*

1. $\mathsf{lfs}(c) \geq \frac{k_1-1}{\sqrt{2}} R$
   Set $k_2 = \frac{k_1-1}{\sqrt{2}} = \frac{\beta-3}{2\sqrt{2}}$
2. $\mathsf{lfs}(q) \geq (k_2 - 1)R = \frac{\beta-3-2\sqrt{2}}{2\sqrt{2}}$

Proof: The center c can not be added to the maximal packing of D due to one of the following reasons:

1. The center c is not in F. Theorem 4.3.6 shows this does not happen.
2. The point-ball of c intersects the point-ball of some point q. Then by Lemma 4.3.8 $\mathsf{lfs}(c) \geq R(\beta-1)/2$
3. The point c intersects the interior of some protective circumscribed sphere $S_1 \in B_1^F$, and then by Lemma 4.3.10 $\mathsf{lfs}(c) \geq R\frac{k_1-1}{\sqrt{2}} = \frac{\beta-3}{2\sqrt{2}}$.
4. c can not be added because it is too close to a 1-polytope $E \subseteq F$. By Lemma 4.3.11 this implies $\mathsf{lfs}(c) \geq \beta R/(2\sqrt{2})$.

The smallest restriction implied by the cases above is by case (3), and hence the result. The other part of the theorem is proved by the Lipschitz condition $\mathsf{lfs}(q) \geq \mathsf{lfs}(c) - R$. □

Lemma 4.3.13 *Let D be a 3-polytope of $\mathcal{B}$, and T one of its tetrahedra generated by algorithm* PACKING *with spacing parameter β. Let ℓ be the length of the shortest edge of T. Let $S = \mathsf{B}(c, R)$ be T's circumsphere. If c intersects the interior of some protective circumscribed face-sphere $S_1 = \mathsf{B}(c_1, R_1) \in B_2^D$ then:*

1. $\mathsf{lfs}(c) \geq R\frac{k_2-1}{\sqrt{2}} = R\frac{\beta-3-2\sqrt{2}}{4}$
2. $\frac{R}{\ell} \leq \frac{\beta\sqrt{2}}{2(k_2-1-\sqrt{2})} = \frac{2\beta}{\beta-7-2\sqrt{2}}$

Where k_2 is a parameter fixed in Lemma 4.3.12

Proof: Identical to Lemma 4.3.10 with k_2 substituted for k_1. □

Lemma 4.3.14 *Let U and V be two intersecting polytopes of dimension ≤ 2. The point-balls of two points p and q placed by algorithm* PACKING *with $\beta > \frac{1}{\sqrt{2}-1}$ on U and V correspondingly are disjoint.*

Proof: The proof is by cases. Assume d_u, the dimension of U, is greater or equal to d_v, the dimension of V. Consider the following cases:

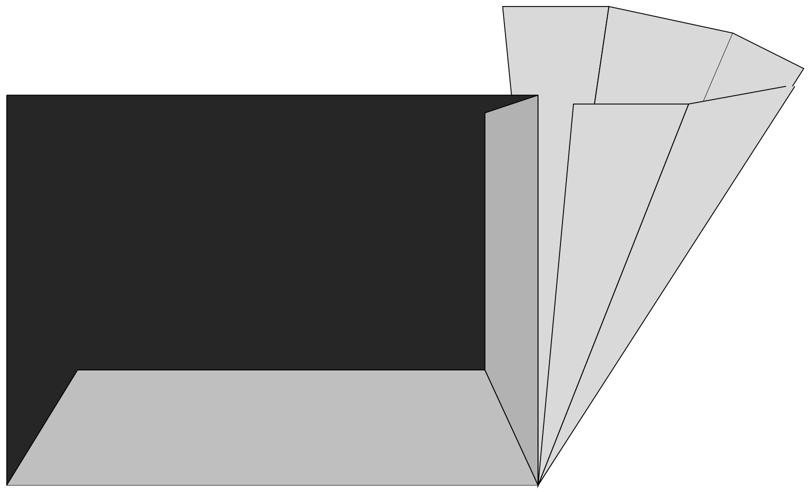


Figure 4.8: Lemma 4.3.14

$d_u = 1$ Then $d_v = 1$ as well, and U and V intersect in a point c. The angle between U and V is at least 90°. It suffices to show that p is below the bisector between U and V. Let $\ell = \|p - c\|$. Recall that c must be a node of the output set, and that c is placed before p. Without loss of generality, assume $\mathsf{lfs}(c) = 1$. The goal is to show that $\mathsf{lfs}(p) \leq \beta\ell/\sqrt{2}$. Since lfs is 1-Lipschitz, $\mathsf{lfs}(c) > \mathsf{lfs}(p) - \ell$. Since p and c co-exist, $\mathsf{lfs}(p) + \mathsf{lfs}(c) < \beta\ell$, or $2\mathsf{lfs}(p) - \ell \leq \beta\ell$, or $\mathsf{lfs}(p) \leq \frac{\beta+1}{2}\ell$. Therefore, if $\frac{\beta+1}{2} \leq \frac{\beta}{\sqrt{2}}$i, p's sphere is below the bisector. This is true whenever $\beta > \frac{1}{\sqrt{2}-1}$.

$d_u = 2$ If $d_v = 1$, the intersection between U and V is a point C. Then, we can argue as before on the two edges (q, c) and (c, p). Therefore, assume $d_v = 2$. Let E be the intersection of U and V. The goal is to show that p is below the bisector plane between U and V. First, assume the perpendicular from p to E meets E. By the algorithm, the sphere of radius $2\mathsf{lfs}(p)/\beta$ at p can at most touch E. Therefore, the radius of p's point-ball at distance ℓ from E is at most $\ell/2 < \ell/\sqrt{2}$ and p is below the bisector.

If the perpendicular to E does not meet E, one of two cases occur. The easier case is that the perpendicular runs into some other edge of U, closer than E, and hence p is still below the bisector. The other case is that V is a plane projecting from interior of the convex face U. U and V are then perpendicular to each other, by the 90° restriction. The part of V closer to q must end in an edge v vertical to U. See Figure 4.8, where U is the horizontal plane, V is the vertical. In that case, the combination of the bisector planes forced by v and E on the points of V, and p's staying far from v and E, ensure the point-balls do not intersect. The edge v forces an additional cone-like protection surface with respect to the plane, as portrayed in the figure.

□

4.4 The number of mesh elements

This section bounds the number of elements in the mesh generated by algorithm PACKING. The technique used follows Ruppert [64].

Theorem 4.4.1 *The number of elements in the mesh constructed by algorithm* PACKING *is smaller than*

$$c \int_{\mathcal{B}} \frac{dV}{\mathsf{lfs}(u)^3}$$

for some positive constant c.

Proof: The mesh nodes are β-spaced according to lfs, hence the balls of radius $\mathsf{lfs}(p)/\beta$ around the nodes of the mesh do not intersect. Let p be a mesh node, u an arbitrary point in p's ball. The distance between u and p is smaller than $\mathsf{lfs}(p)/\beta$, therefore by the Lipschitz condition of lfs:

$$\mathsf{lfs}(p) - \mathsf{lfs}(p)/\beta \leq \mathsf{lfs}(u) \leq \mathsf{lfs}(p) + \mathsf{lfs}(p)/\beta.$$

Hence,

$$\mathsf{lfs}(p)(1 - \frac{1}{\beta}) \leq \mathsf{lfs}(u) \leq \mathsf{lfs}(p)(1 + \frac{1}{\beta}).$$

Therefore the integral over the ball contributes a constant to the total integral. □

Further following Ruppert, it can also be shown that any bounded aspect ratio mesh accommodating $\mathcal{B}$ must have at least $\Omega(\int_{\mathcal{B}} \frac{dV}{\mathsf{lfs}(u)^3})$ elements. Note however that the meshes we generate are not of bounded aspect ratio, thus this is not an optimality result in the usual sense; rather, what we have shown is only that the bounded radius-edge ratio mesh we generate contain about as many elements as a bounded aspect ratio mesh conforming to the same domain would.

A plausible application of generating bounded radius-edge ratio meshes is as a first step towards generating a bounded aspect ratio mesh. In the next step, slivers elimination is attempted, for example by using a recent algorithm by Chew [22]. Eliminating all slivers transforms the mesh from a bounded radius-edge ratio mesh into a bounded aspect ratio mesh. With that application in mind, it is reasonable to compare the number of elements in our meshes to that of any other bounded aspect ratio mesh conforming to the same domain.

Nonetheless, this result is not optimal. In particular, there are examples of domain descriptions with a very small lfs value, that can be tetrahedralized using a few large slivers. Consider Figure 4.9. Two segments crossing close to each other induce a small feature, but can be tetrahedralized using one sliver.

Clearly, a different local spacing function is necessary to capture the optimal node spacing of bounded radius-edge ratio meshes. Finding this function is currently still an open problem. The following examples demonstrate the difficulties:

Figure 4.9 indicates that if two segments pass close to each other and yet do not touch, the local spacing must be large and fixed only by the segments' end points. However, what should the spacing be when three segments all pass close to the same location? that depends on the arrangement of the segments. Consider three line segments a, b and c. Let a and b be stacked upon each other as in Figure 4.9. The right local spacing depends on where segment c penetrates the convex hull of a potential sliver. See Figure 4.10: changing the angle c forms with the other two segments changes the local spacing. For example, imagine c projecting towards you from the picture plane and yet passing close to the point where a and b are closest, the local spacing must then be arbitrarily small since c would then penetrate the sliver formed by a and b. Therefore, the local spacing depends on the

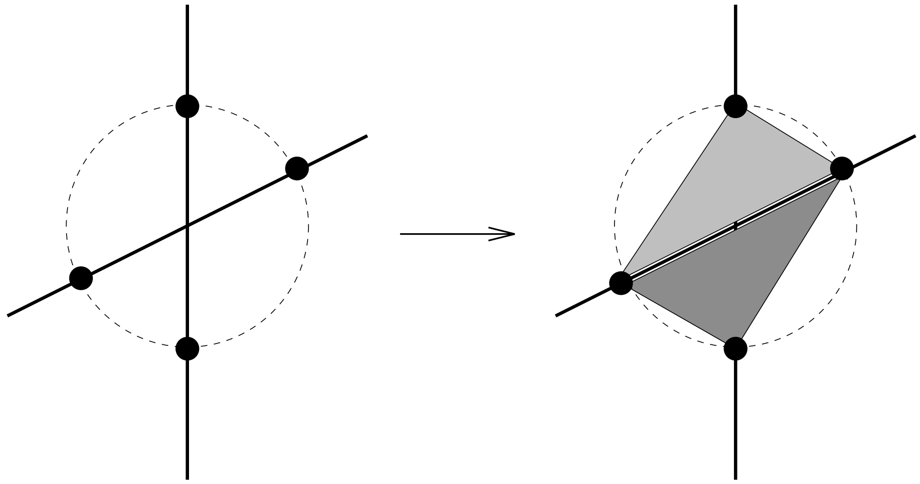


Figure 4.9: On the left, two boundary segments pass close to each other but do not intersect. The vertical segment is behind the horizontal segment. This arrangement can be triangulated using a sliver in the area where the two segments approach each other: draw a sphere that serves as the target sliver circumscribed sphere, and place nodes at the intersection of the segments and the sphere. The resulting tetrahedron, on the right, is a sliver conforming to both segments. The edge lengths of the sliver can be much larger than the edge lengths of bounded aspect ratio tetrahedra conforming to the same domain. The closer the segments pass to each other, the larger the number of bounded aspect ratio tetrahedra necessary to triangulate the arrangement, but the number of bounded radius-edge ratio tetrahedra remains constant.

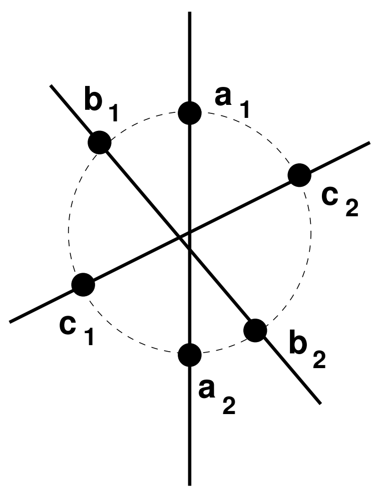


Figure 4.10: Three line segments, a, b and c all pass close to the same point. Imagine them stacked on top of each other, with c closest to the viewer. Also, c does not penetrate the convex-hull of $\{a_1, a_2, b_1, b_2\}$. The convex-hull of all six nodes can be decomposed into four slivers:$\{[a_1, a_2, b_1, b_2], [b_1, b_2, c_1, c_2], [a_1, b_1, b_2, c_2], [a_2, b_1, b_2, c_1]\}$. However, if segment c is rotated so it is perpendicular to the picture plane, the local spacing becomes much smaller and a few slivers no longer suffice to decompose the convex hull.

relative positions of all three segments. Similar observations can be made for sets of four and more segments that all pass very near to one location.

4.5 Summary

This chapter concentrated on presenting an algorithm to generate a bounded-aspect ratio Delaunay mesh conforming to a given piecewise linear non-manifold domain. The algorithm proceeds by packing points on the different boundary elements in increasing order of dimension and finally in the interior. Points were placed on a boundary element or an interior cell until no more points could be added, but this maximality condition can be relaxed to require only Delaunay maximality: that is, only Delaunay circumcenters have to be added if they can be added without violating any restriction. Other possible relaxations of the definition of maximality are possible but can not be directly derived from the theorems in this chapter the way Delaunay maximality can. An example for such a relaxation is requiring that if a small ϵ-sphere of points can be added, at least one of them must be. For $\epsilon = 0$ the old definition of maximality is obtained. Different ways of relaxing the notion of maximality lead to different algorithms and in that sense this chapter supplies the foundations for proving the radius-edge ratio condition for different methods of laying down the mesh nodes.

Part II

Mesh Coarsening

Chapter 5

Mesh Coarsening

Numerical methods such as the finite element, finite difference, and finite volume methods apply the following basic steps to solve a partial differential equation (PDE) over a continuous domain Ω:

1. **Geometric modeling:** the continuous domain is approximated using a simpler, discrete description.

2. **Mesh generation:** the interior of the domain is decomposed into a mesh M of simple and well-shaped elements.

3. **Approximation:** a system of linear or non-linear equations is formed over M for the governing PDEs (e.g., the stiffness matrix and the right hand vector are assembled).

4. **Solution:** the system of equations is solved, and the error of the solution estimated.

5. **Adaptive refinement:** if necessary, the mesh is refined and steps 3 and 4 will be repeated over the refined meshes.

Once the mesh M is generated, a system of equations is defined over M and has to be solved. The class of multi-level techniques has become one of the most effective and successful classes of numerical techniques for solving PDEs. These techniques have been used in multigrid methods [12] and multi-level domain decomposition [15]. A multi-level method solves the system by first constructing a *coarsening sequence* of meshes $M_0, \ldots, M_k$, where $M_0 = M$ is the finest mesh that discretizes Ω. For each i in the range $1 \leq i \leq k$, the mesh M_i is a geometric coarsening of M_{i-1}. The essence of multigrid methods and other multi-level numerical methods is the transformation of partial solutions from mesh M_i to mesh M_{i-1} using interpolation, and from mesh M_i to mesh M_{i+1} using restriction. Informally, these methods solve a PDE on Ω by first obtaining an initial solution either for M_0 or for M_k, and then improving the quality of the solution by transforming it up and down the coarsening sequence hierarchy while applying some simple and efficient iterative methods at each level.

The simplest form of a coarsening sequence is a series of nested structured meshes, known as regular grids. For this class of meshes Brandt [12] showed, by carefully using restriction and interpolation, that the solution for M_0 can be efficiently obtained using multigrid methods. See also Bramble, Pasciak and Xu [11]. Nested structured coarsening sequences are attractive choices in practice because they can be easily generated and because the convergence behavior of the structured multigrid methods on such sequences is well understood. However, the use of structured regular grids limits the applicability of the method to problems whose domains are simple and whose solution functions have small or constant Hessian [49, 15, 71].

The use of unstructured meshes is inevitable in the solution of complex problems with more intricate domain geometry and solutions. The theory behind applying multigrid and multi-level methods to unstructured meshes is not as well understood as it is for the structured case. However, it is an area of active research, with increasing theoretical progress [11, 15, 17]. In practice, multigrid methods on unstructured meshes are popular despite this current lack of full theoretical foundations [6, 57].

The algorithmic and computational geometry aspects of generating coarsening sequences are important issues to address in view of the theoretical and experimental interest in applying multigrid and multi-level methods to unstructured meshes. The effectiveness of a multi-level method that uses an unstructured coarsening sequence $M_0, \ldots, M_k$ depends on the quality of this sequence [11, 15, 17]. In particular, Chan and Zou [17] provided sufficient conditions for multi-level additive Schwarz methods to work on unstructured meshes. Informally, their conditions require, for each i in the range $1 \leq i \leq k$, that (1) M_i is well-shaped, e.g., in two dimensions elements of M_i should have a bounded aspect ratio; and (2) M_i approximates M_{i-1} in the numerical formulation. The coarsening problem can be informally defined as follows: Given a well-shaped mesh M_0 and a threshold size b, construct a sequence $M_1, ...M_k$ with $|M_k| = O(b)$ that satisfies conditions (1) and (2).

The problem of mesh coarsening can be viewed as a generalization of the problem of mesh generation. In particular, in mesh generation a single mesh is constructed whereas in mesh coarsening a sequence of meshes has to be constructed. Furthermore, the coarsener is restricted to sample only from the discrete set of the initial mesh nodes instead of having the freedom to sample arbitrarily from the domain. Thus, the challenges of mesh coarsening, as contrasted with mesh generation, are twofold: first, that of understanding the shape and size of each mesh in an optimal sequence that satisfies conditions (1) and (2), and second, the challenge of constructing such a sequence over a discrete node set. However, without the foundations laid by recent advances in the theoretical understanding of mesh generation techniques, as in the work of Bern, Eppstein and Gilbert [9], Chew [21], Ruppert [64], and Mitchell and Vavasis [52], this work would be impossible.

This chapter describes a new algorithm for the coarsening problem in two dimensions. This algorithm guarantees that the coarsening sequence satisfies conditions (1) and (2). It also minimizes the size of the mesh at each level up to a constant factor. A sequence is *node-nested* if the set of the nodes of M_i is a subset of that of M_{i-1}. The approach presented here can be used to generate both node-nested and non-nested coarsening sequences.

The chapter is structured as follows: Section 5.1 defines the problem of mesh coarsening. Section 5.2 reviews previous work on mesh coarsening. Section 5.3 presents the new coarsening approach. The rest of the sections analyze the correctness of this approach, whereas Section 5.8 discusses issues of efficiency and presents a version of the algorithm that has a linear complexity run time.

5.1 The problem of mesh coarsening

Definition 5.1.1 (A coarsening) *A **coarsening** $M' = (\mathcal{P}', E', \mathcal{B})$ of a mesh $M = (\mathcal{P}, E, \mathcal{B})$ is a mesh whose edge length function $\mathsf{el}_{M'}$ is point-wise larger than el_M but still conforms to the same domain. The mesh M is then a **refining** of M'.*

Mesh coarsening is the inverse of mesh refining. Mesh refinement is employed when adaptively computing numerical quantities. In areas of the mesh where more accurate information is desired, or where large numerical errors are suspected, the mesh is refined. For some of the rich literature on mesh

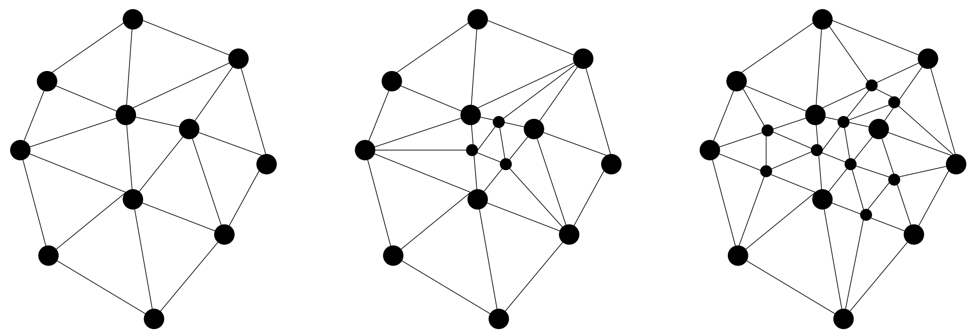


Figure 5.1: **Regular Refinement** is a popular method to refine a mesh [4, 3]: the element to be refined is partitioned into four similar smaller triangles by adding new nodes at the middle of the edges of the elements, and connecting the new nodes. In the figure, the original nodes are drawn larger. This form of refinement has the advantage that the four new triangles do not introduce new angles and therefore do not degrade the quality of the mesh. Note however that a triangle that neighbors a refined triangle must be partitioned as well, by connecting the new node on its edge to the node opposite to it. This does introduce new, smaller angles. In order to prevent repeated degradation, triangles with only one or two new nodes on their edges are never refined again. Rather, if that triangle is to be refined, the previous partition is undone and nodes are placed on all the edges, yielding the standard good refining. Thus, this form of refining uses auxiliary temporary refinings. The implication of the reversal of the temporary refining is that the method does not yield a nested coarsening sequence.

refinement, see [62, 4, 41, 53] and the two excellent surveys [10, 8]. These papers present a variety of refinement techniques, such as longest edge bisection [70, 62, 63], newest vertex bisection [53, 54], vertex ordered simplicial bisection [44] and regular refinement [4, 3].

If all intermediate meshes have been kept, mesh coarsening can be obtained by undoing the changes made, reversing the refinement process. However, it is unreasonable to assume that such a mesh refinement history is always available, or if available, that it generates an optimal coarsening sequence. In a black-box approach to mesh coarsening, the only information available is the input mesh M. Additionally, one may not assume the mesh was derived by a particular refinement method.

The coarsening can be **element-nested**, **node-nested** or **non-nested**. In an element-nested coarsening, each element of M' can be represented as a union of elements in its refinement M, while elements in the other two types may be unrelated. A coarsening is **node-nested** if each node of mesh M' is also a node of mesh M; otherwise it is **non-nested**.

In general, a triangular mesh does not have any element-nested coarsening, unless it was carefully crafted as such. Even mesh refinement methods that mostly use element decomposition in the refinement process are not purely decomposition based; a certain amount of edge flipping and deleting is necessary to maintain good aspect ratio, as the example of Figure 5.1 shows.

Coarsening, even in the relaxed sense of **node-nested** meshes, might cause a degradation in the aspect ratio of the coarser mesh compared to that the finer one. This should be avoided. The goal is that of automatically generating a sequence of coarsenings from an initial high-quality mesh, with tight controls over the aspect ratio and the number of elements of each coarsening mesh, and on the number of meshes in the sequence.

Definition 5.1.2 (Coarsening Sequence) *A coarsening sequence of a mesh M_0 is a sequence of meshes $\mathcal{M} = M_0, M_1 \ldots M_k$, such that M_{i+1} is a coarsening of mesh M_i.*

Definition 5.1.3 (Length and width of a coarsening sequence) *The length of a coarsening sequence is the number of meshes (levels) in the sequence; the width of the sequence is a function of its level: at level i, the width is $|M_i|$, the number of nodes in mesh M_i.*

Definition 5.1.4 (Good aspect ratio coarsening sequence) *Let θ be a constant, fixed angle; let b be a positive integer. A coarsening sequence $\mathcal{M}$ is (θ, b)-**good aspect ratio** coarsening sequence if $|M_k| = O(b)$ and θ is a bound on the smallest angle of $M_i \quad 0 \leq i \leq k$.*

The coarsening sequence meshes should approximate each other well. This can be expressed via their edge-length functions, or by requiring that an element of a coarse mesh intersect a small number of elements of the preceding finer mesh. For a good aspect ratio mesh, these two definitions are equivalent.

Definition 5.1.5 (Local similarity) *Two meshes M_1 and M_2 are $\mathcal{C}$-**locally similar** if $\frac{1}{\mathcal{C}}\mathsf{el}_{M_2} \leq \mathsf{el}_{M_1} \leq \mathcal{C}\mathsf{el}_{M_2}$.*

Definition 5.1.6 (Bounded local intersection) *Two meshes M_1 and M_2 are of local intersection bounded by a constant $\mathcal{I}$ if each element of M_1 intersects at most $\mathcal{I}$ elements of M_2 and vice-versa.*

Lemma 5.1.7 *Let M_1 and M_2 be two meshes with aspect ratio bound θ. If they are locally similar, they have the bounded intersection property and vice versa.*

Proof sketch: First, assume the meshes are locally similar. Look at triangle T_1 of M_1. Local similarity implies that each triangle T_2 of M_2 that intersects T_1 has similar maximal edge length. Since all triangles are of bounded aspect ratio, a volume argument shows that there can be no more than a constant C such triangles intersecting T_1. Note that C must depend of the aspect ratio of both M_1 and M_2.

Now, assume that the the meshes have the bounded intersection property. If triangle T_1 of M_1 intersects a very small triangle T_2 of M_2, than it must also intersect many of its neighbors. The smaller T_2 is relative to T_1, the larger the number of T_2's neighbors that T_1 must intersect. Since the number of triangles T_1 intersects is bounded by a constant, the size of T_2 relative to T_1 must also be bounded by a constant. M_1 and M_2 are therefore locally similar. □

Definition 5.1.8 (Well-shaped coarsening sequence) *A coarsening sequence $\mathcal{M} = \{M_0 \cdots M_k\}$ is $(\theta, \mathcal{C}, b)$-well-shaped if*

- *$\mathcal{M}$ is a good aspect ratio coarsening sequence with bound θ.*
- *Mesh M_i and M_{i+1} are $\mathcal{C}$-locally similar.*
- *$|M_k| = O(b)$*

The first condition, good aspect ratio, is motivated by iterative methods: At each level of the multigrid method an iterative method is used for a few rounds to "smooth" the error. The convergence properties of iterative methods are related to the aspect ratio of the underlying mesh. The second condition, local similarity, is motivated by the restriction and interpolation phases of the multigrid methods, used to transform partial solution between meshes in adjacent levels of the hierarchy. To reduce the interpolation and restriction errors, adjacent meshes should approximate each other well, i.e. be locally similar.

Definition 5.1.9 (The problem of mesh coarsening) *Given M_0 and b, the problem of* ***mesh coarsening*** *is to find a $(\theta, \mathcal{C}, b)$-well-shaped coarsening sequence, with constant bounds θ and $\mathcal{C}$ on the aspect ratio and local similarity of the mesh sequence, and with smallest (up to a constant factor) depth and width.*

5.2 Previous work

Mesh coarsening is an active area of research. Similar to mesh generation, it has two aspects: numerical analysis and computational geometry aspects. Most of the previous researches have concentrated on the numerical analysis aspect of mesh coarsening, which is important since the multigrid method for unstructured meshes is in its development phase, and the theory behind it not fully understood yet. In addition, numerical analysis investigation also helps to uncover the necessary and sufficient geometric conditions that a desirable coarsening sequence should satisfy. Nonetheless, as the multigrid method has been gaining some acceptance and maturity, its algorithms and software development require better understanding of the underlying computational geometry issues. Unfortunately, to the best of our knowledge, all current algorithms for mesh coarsening come with no guarantees for the size and shape of the elements they produce. In fact, in most cases we can provide counter-examples that show these algorithms can fail to produce a coarsening sequence with such guarantees. The algorithm developed in this paper is the first provably good mesh coarsening algorithm. One objective of this this work is to initiate the study of mesh coarsening from its computational geometry aspects.

Previous approaches to mesh coarsening can be classified into the following categories:

5.2.1 Topological coarsening

Arguably the most popular paradigm. The topological structure of the mesh, i.e., the neighborhood graph structure of either its elements or nodes, is used. The two prominent topological based methods are MIS-based coarsening and volume agglomeration coarsening.

MIS-based methods

MIS-based coarsening uses the 1D skeleton of the mesh, i.e. the neighborhood graph of its nodes. A subset of the mesh nodes which is a maximal independent set of the 1D skeleton, is selected to be the coarser mesh nodes. This node set is then triangulated (e.g. by a Delaunay triangulation), and the process repeated. The coarsening sequence is created iteratively, from the finest mesh to the coarsest.

Definition 5.2.1 (Maximal Independent Set (MIS)) *The MIS of a graph $G = (V, E)$, where E is the edge set and V is the node set, is a set $U \subseteq V$ such that the following are true:*

1. ***independence:*** *if $u_1, u_2 \in U$ then $(u_1, u_2) \notin E$, and*
2. ***maximality:*** *no node $v \in V$ can be added to U without violating independence, i.e. all nodes $v \in V \setminus U$ are incident to some node $u \in U$.*

Guillard [34] suggests a combination of Delaunay based triangulation with an MIS-based coarsening scheme. Nodes on the input PSLG (the boundary) segments are at higher priority, and are considered first for inclusion in the MIS. The boundary nodes are further prioritized by their boundary angles, so that nodes incident to sharper angles have a better chance at surviving the coarsening.

Procedure: MIS_COARSENING(M_0)

1. Let k be the length of the required coarsening sequence.
2. For $i = 1$ to k
   - Let G_{i-1} be the 1D skeleton of M_{i-1}.
   - Find X_i, a maximal independent set of G_{i-1}.
   - Construct M_i by triangulating X_i.

   End For
3. Return $(M_1, ..., M_k)$.

Figure 5.2: Basic scheme of maximal independent set coarsening

Ollivier-Gooch [57] also proposes an MIS-based scheme. The boundary nodes are again considered first. His method tries to preserve important geometric features, typically sharp corners. Furthermore, at least three nodes are retained from each closed boundary curve. In an attempt to increase the size of the MIS, vertices that are selected can be exchanged with their unselected neighbors. The re-triangulation phase is done incrementally via edge contraction: as a node is removed, one of its neighbors is selected and the edge between them is contracted. Finally, edge swapping is employed to locally minimize the maximal angle.

Dorward, Matheson and Tarjan [26] discuss two strategies to generate the coarse mesh nodes, which are related to the MIS. The first, **dominating sets**, is a superset of a maximal independent set, i.e. a set of nodes is selected such that all the nodes removed neighbor a node of the dominating set. The second approach, **independent set**, removes a maximal independent set from the mesh and retriangulates the remaining nodes. (This notation might be a little confusing, since the nodes they retain are the inverse of the maximal independent set). They pay attention to the problem of obtaining a dominating set with at most $1/3n$ nodes where n is the number of nodes in the fine mesh, i.e. they seek to minimize the size of the coarser mesh. They provide size guarantees for the resulting mesh, but no aspect ratio guarantees.

Volume agglomeration

This topological method is applied to control-volume discretizations. Each mesh node is assigned a volume; this collection of volumes is the dual of the mesh. Two popular constructions for the dual are the Voronoi diagram (when the mesh is a Delaunay triangulation), or the volume formed by connecting all the midpoints of edges and centers of triangles incident to a node.

Working from the dual mesh, agglomeration methods coarsen the mesh by joining neighboring volumes together. Hence, the complexity of the cell or volume description increases as the mesh is coarsened. The coarsening proceeds by iteratively considering all volumes. If a volume has been coarsened already, it is skipped. Otherwise, the volume grabs some or all of its un-coarsened neighboring volumes, to form a new larger cell. As in the MIS-based schemes, the order by which the volumes are traversed can be prioritized to proceed from the boundary cells inward, or in an advanced front fashion. See [72, 48, 46, 40].

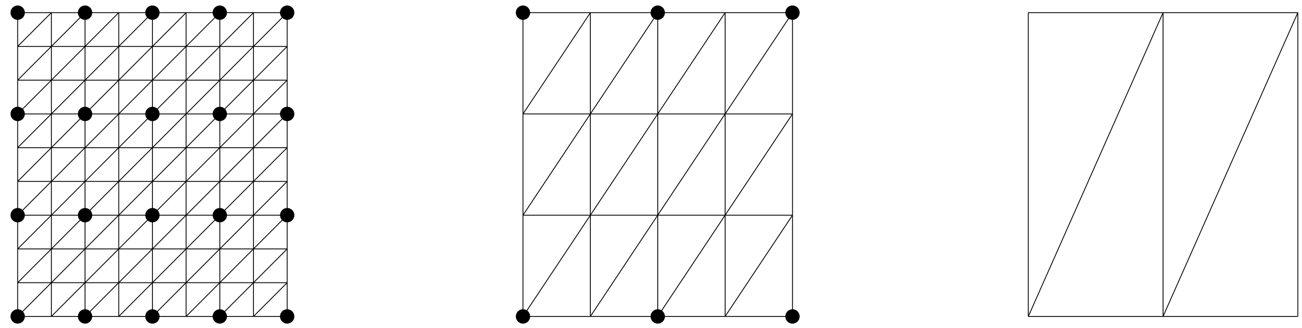


Figure 5.3: Repeated applications of MIS can degrade the aspect ratio.

5.2.2 Repeated degradation: problems of topological coarsening

MIS on the 1D skeleton 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 coarsening sequence, such as its aspect ratio. The problem is illustrated in Figure 5.3: 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 very uniform meshes, as in the grid-like mesh of Figure 5.3. Agglomeration meshes are dual to MIS-based meshes, and as such the geometric shape of the cells is subject to the same problems.

For general unstructured meshes, the problem of repeated degradation is much more severe. In particular, the following lemma is true:

Lemma 5.2.2 *There exists an unstructured mesh such that all possible choices of MIS result in a coarsening sequence with increasingly worse aspect ratios.*

The proof is by construction. Let M_0 be a one dimensional mesh whose nodes are $P = \{2^{i-1} - 1 : i = 1 \cdots n\}$. The edges of the mesh are between adjacent nodes, M_0 is a line graph of n nodes. The **aspect ratio** of a one dimensional mesh is the maximum ratio between two adjacent edges, hence the aspect ratio of M_0 is equal to 2.

For this mesh there is no well-shaped coarsening sequence of depth $\log n$. To prove this statement we need some of the results of latter sections, which prove the following statement: any coarsening sequence of this mesh that maintains both a good aspect ratio and the local similarity property is similar (again, in the sense of local similarity) to the coarsening sequence formed by repeatedly coarsening the smallest edge only. The optimal coarsening can reduce the mesh size by a constant number of elements only. Since topological coarsening methods always reduce the size of the mesh by a constant factor at each level, they must fail to produce good aspect ratio coarsenings.

To gain more intuition into the problem, look at a very natural choice of a coarsening sequence where the jth mesh $\{M_j\}$ has the following node set: $P_j = \{2^{i2^j-1} - 1 : i = 1 \cdots n/2^j\}$. In other words, P_j is formed by taking every other point of $\mathcal{P}_{j-1}$. The aspect ratio of mesh M_j is therefore at least $2^{2^j} - 1$, and worsens super-exponentially. Contrast that with the quasi-uniform case, where a rather deliberate bad choice of a coarsening sequence was necessary to exhibit a pattern of repeated degradation.

5.2.3 Derefinement methods

The mesh coarsening sequence is obtained by reversing a refinement process of the mesh. This entails storing the intermediate meshes during the refinement process. It also assumes these meshes will constitute a good coarsening sequence. This reasonable assumption is not always true: whereas the element shape can be trusted to be of high quality, it is not clear that the sizes of the meshes thus derived will decrease optimally. The second property depends on how aggressively was the mesh refined.

This method has clear advantages: the fine elements could be constructed to be almost always nested in the coarser elements. The theory of multigrid for nested elements is better understood. However, this method is not viable for a black box approach.

Of interest is the hybrid method suggested by Bank and Xu [5]. They try to implement a derefinement method without the intermediate information. The input fine mesh is assumed to have been constructed by a nested elements refinement process, which they aim to reverse. Using a heuristical approach, elements are picked and attempted to be coarsened out. When the process gets stuck, edge flipping is used.

5.2.4 Multi-level mesh construction

This refers to a general framework, where a sequence of coarse meshes is generated under the sole constraint that they approximately fit the same boundary. The different meshes are totally unrelated and the finite element spaces are non-nested. In essence, mesh generation methods are employed at each level over a given boundary description. This method can be seen as a general framework for generating a non node-nested coarsening sequence given access to a detailed domain description.

5.2.5 Other coarsening methods

Barth [6] suggests a coarsening framework he terms "randomized multigrid". Given the initial mesh, the coarsest mesh is first constructed. Then, using a Ruppert-like [64] incremental mesh approach, the fine mesh nodes are added one by one. The node insertion is prioritized by the scope of the Delaunay triangles in the current mesh. The *scope* is the number of fine mesh nodes that are inside the Delaunay circumsphere. The node closest to the center of the highest scope triangle is inserted into the mesh.

Momentarily ignoring the largest-scope rule, this is a general framework for a coarse-to-fine coarsening sequence construction. Furthermore, Barth points that a randomized insertion order allows using previous results about randomized incremental Delaunay algorithms and data structures [33], that make computing the prolongation/restriction operators simple and efficient.

The "largest-scope first" rule contradicts the random ordering, but Barth presents experimental results that show that over his test-suite this did not impair efficiency. However, from a geometrical point of view, the highest-scope rule can lead to non-optimal coarsening sequences, especially in graded unstructured meshes. Consider for example the case where the mesh has a convergence point. The mesh must be refined in the larger features areas at the same time it is refined in the smallest areas if the coarsening sequence is to be of good aspect ratio, but this does not always occur under the largest-scope rule. Clearly, additional rules giving priority to refining bad aspect ratio triangles are necessary.

5.3 New paradigm: function-based coarsening

In this section, a new paradigm for mesh coarsening is introduced. This paradigm uses a representation termed "spacing functions" to express the geometric structure of the mesh. This representation can be coarsened without the pitfall of repeated degradation. The spacing functions, along with the point set of the initial mesh, capture all the information necessary for the provably good generation of the coarsening sequence. This method relies heavily on the spacing function definitions, properties and techniques that were introduced in Chapter 2.

At a high level, M_0 defines a spacing function f_0 which describes the typical size and point spacing of M_0: f_0 is defined by its values at the mesh nodes. The idea is to compute a spacing function f_i for each level, and use it to generate M_i from the original (or previous) point set. Given the spacing functions the task is then to create a point set that is "spaced" according to that function, and triangulate it. We call this coarsening technique **function-based coarsening**. It contains four steps:

1. recover the spacing function of the initial mesh;
2. increase the spacing value of the mesh points smoothly to obtain the new spacing functions;
3. delete some mesh nodes so that the remaining nodes are spaced according to the new spacing function; and
4. compute the Delaunay triangulation of the nodes obtained in Step 3.

The function-based coarsening approach is presented here with the assumption that all the meshes of the coarsening sequence conform to the same boundary description. Thus, coarsening is not attempted beyond the demands of the boundary description, i.e., coarsening is stopped when all the remaining nodes are necessary to guarantee the existence of a good aspect ratio mesh conforming to the boundary. It is sometimes desirable to combine mesh coarsening with boundary description simplification. In mesh coarsening, boundary nodes lying on the segments may be removed but the boundary segments end-points are always retained. In boundary simplification, some of the boundary segments end-points may be removed and a new boundary description is formed, under the restriction that the new boundary approximates the old boundary. This might enable continuing the coarsening beyond the original boundary restrictions. However, it is easy to find domain counter-examples where no reasonable boundary simplification that approximates the original boundary allows for a good aspect ratio meshing. The problems of the conflicting goals of boundary simplification and mesh coarsening are discussed in detail in Section 7.4.

Nonetheless, for many interesting and practical domains the ideas of function based-coarsening can be easily adapted to handle simplification as well. Figure 5.4 shows function-based coarsening employed on a complex boundary, where the boundary is allowed to be coarsened.

The basic coarsening scheme is described in algorithm ONE_LEVEL_COARSEN of Figure 5.6. The following subsections explain in more detail the different steps of the algorithm.

5.3.1 Recovering the spacing function

The initial spacing function is recovered from the natural spacing of the original mesh (this following function and other definitions relating to spacing functions appeared in Section 2.2):

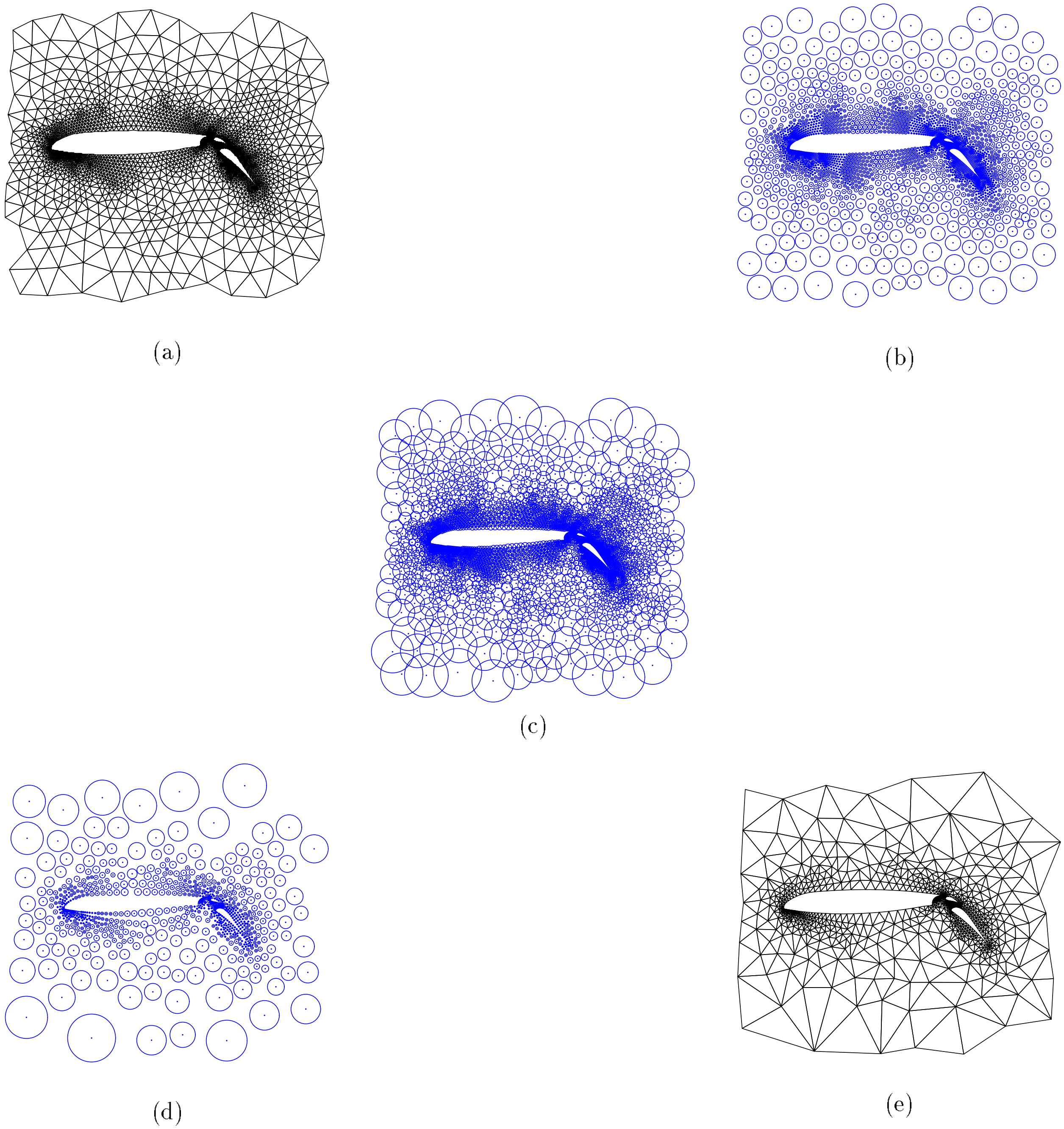


Figure 5.4: Function-based coarsening: the spacing function at a node is denoted by a ball whose radius is the function value. The initial mesh is in (a). The first step is recovering the spacing function of the mesh using the function 0.5NN (b). The function is then coarsened (c), and a subset of the nodes forming a maximal set of disjoint balls is picked (d) and triangulated (e).

Definition 5.3.1 *The **nearest neighbor** (NN) function of a point set $\mathcal{P} \subset \Omega$ assigns to each point $p \in \mathcal{P}$ the distance to the point $q \in \mathcal{P}$ that is nearest to it. It can be extended to the domain Ω by assigning to a point $x \in \Omega$ the radius of the smallest closed ball centered at x and containing at least two points from $\mathcal{P}$.*

Remark: This function is related to the lfs function suggested by Ruppert [64] (see Definition 2.2.5 of Section 2.2). $\mathsf{lfs}_{\mathcal{B}}$ depends on the domain description, whereas $\mathsf{NN}_{\mathcal{P}}$ depends on the point set of an actual discretization of the domain. Note that $\mathsf{NN}_{\mathcal{P}}$ can be much smaller than $\mathsf{lfs}_{\mathcal{B}}$, but Theorem 2.2.8 implies it can be at most a constant factor larger for a good aspect ratio mesh.

In essence, if the boundary was so simple as to be just a collection of points $\mathcal{P}$, then (by the definition of the respective functions) $\mathsf{NN}_{\mathcal{P}} = \mathsf{lfs}_{\mathcal{P}}$. However, in the coarsening framework where the boundary description is distinct and different from the mesh nodes, the mesh can be much finer than its domain dictates. Since Ruppert showed that a good aspect ratio mesh must conform to the $\mathsf{lfs}_{\mathcal{B}}$ function, coarsening can proceed only as long as $\mathsf{NN}_{\mathcal{P}} < c\mathsf{lfs}_{\mathcal{B}}$. The constant c depends on the desired aspect ratios of the coarsened meshes.

5.3.2 Coarsening the spacing functions

The spacing function is coarsened by first obtaining the local coarsening restrictions, and then generating a global spacing function answering to all these restrictions. In particular, each node has a desirable coarsening goal which also gives an upper bound for the coarsened function values at other nodes because of the Lipschitz condition. The final coarsening function is the largest Lipschitz function that obeys all the upper bounds at all the nodes.

The following definition describes this coarsening function when the goal is to coarsen the spacing function by some arbitrary (and not necessarily a constant) coarsening factor $\mathcal{C}$:

Definition 5.3.2 (Coarsening function) *Let $\mathcal{P}$ be a point set in a domain $\Omega \subseteq \mathbb{R}^2$. Let g be a spacing function over Ω. Let $\mathcal{C} > 1$ be an arbitrary real number. The **local coarsening** of f at point $q \in \mathcal{P}$ is a cone-shaped function:*

$$\hat{f}_{g,\mathcal{C},q}(x) = \mathcal{C} \cdot g(q) + \|q - x\|$$

*The $\mathcal{C}$-**coarsening** of f with respect to $\mathcal{P}$ is defined as a minimum of all the local coarsenings:*

$$f_{g,\mathcal{C},\mathcal{P}}(x) = \min_{q \in \mathcal{P}} \hat{f}_{g,\mathcal{C},q}(x),$$

When clear from the context, g and $\mathcal{P}$ are omitted from the notation. For complex domains, the distance metric used is the geodesic one.

The local coarsening function looks like a cone, lifted from the plane to a height equal to $\mathcal{C}$ times the previous function value. The coarsening function looks like the lower envelope of all the cones associated with the local coarsening functions. Figure 5.5 shows a cone, and intersection of cones.

For quasi-uniform meshes, a simpler coarsening function suffices:

Definition 5.3.3 (Threshold Coarsening function) *Let $\mathcal{P}$ be a point set in a domain $\Omega \subseteq \mathbb{R}^2$. Let g be a spacing function over Ω. Let $\mathcal{C} > 1$ be a real number. The C-**threshold-coarsening** function is defined as:*

$$t_{g,\mathcal{C},\mathcal{P}}(x) = \max(g(x), \mathcal{C} \min_{y \in \Omega} g(y))$$

(a) (b)

Figure 5.5: (a) The local coarsening function of point p is a cone centered at p, and lifted above p by $\mathcal{C}f(p)$. (b) The global coarsening function is the lower envelope of the collection of cones associated with the points.

The threshold coarsening function coarsens all features smaller than $\mathcal{C}\min g$ *but does not affect larger features.*

5.3.3 Coarsening the meshes

Let M_0 be the initial mesh, $\mathcal{C}$ the factor by which the mesh should be coarsened. Refer to Figure 5.6 for the basic algorithm. To coarsen the mesh, first generate the coarsened spacing function values for each node of the mesh using C, and then pick a subset of the mesh nodes which is β-spaced by this spacing function. One possible method to pick the coarsened mesh nodes is by using a conflict graph.

Definition 5.3.4 *The* ***conflict graph*** *of a point set* $\mathcal{P}$, $CG(\mathcal{P})$, *with respect to a* β*-spacing function* f *and a boundary description* $\mathcal{B}$, *is a graph* $CG(\mathcal{P}) = (\mathcal{P}, E)$ *where*

$$E = \left\{ (p_i, p_j) : \|p_i - p_j\| < \frac{f(p_i) + f(p_j)}{\beta} \right\}$$

such that the segment (p_i, p_j) *is contained in the domain. (node* p_i *visible to node* p_j*).*

The coarsened point-set is essentially a MIS of the conflict graph with respect to the coarsened spacing function. This basic scheme is enhanced in order to handle the boundary. The coarse point set, $\mathcal{Q}$, is built in increasing order of dimensions: initially the boundary segment end-points and points are included in $\mathcal{Q}$, then a MIS of the conflict-graph of the boundary nodes is added to $\mathcal{Q}$, new boundary edges are formed by edge-contraction of removed boundary nodes, and finally, $\mathcal{Q}$ is completed to include the interior nodes that are not too close to the boundary edges, and that form an MIS of the corresponding conflict-graph.

To guard the boundary edges, a protective zone around each edge is used:

Definition 5.3.5 (γ protective zone) *Let* e *be an edge. The* γ *protective zone around* e *is a rectangle of height* $2\gamma|e|$, *extending to a height of* $\gamma|e|$ *on each side of* e *and of width* $|e|$. *If this*

Procedure: ONE_LEVEL_COARSEN$(M_0, f_0, \mathcal{C}, \beta, \gamma)$

Input: $M_0 = (\mathcal{P}, E, \mathcal{B})$, a mesh over a domain described by $\mathcal{B}$.
f_0, the spacing function of M_0.
$\mathcal{C}$, the coarsening factor.
β, the spacing constant.
γ, the boundary edges protective zone constant.

Output: $M_1 = (\mathcal{Q}, E_1, \mathcal{B})$, a coarser triangular mesh conforming to $\mathcal{B}$.

Method:

1. Compute $f_1(p_i) = f_{f_0,\mathcal{C},\mathcal{P}}(p_i)$ for each $p_i \in \mathcal{P}$.
2. Let $\mathcal{P}^0$ be the boundary points; $\mathcal{P}^1$ M_0's nodes that are located on the segments of $\mathcal{B}$; $\mathcal{P}^2$ the mesh nodes. Note that $\mathcal{P}^0 \subseteq \mathcal{P}^1 \subseteq \mathcal{P}^2 = \mathcal{P}$.
3. Set $\mathcal{Q}^0 = \mathcal{P}^0$.
4. Let $G_1 = \mathsf{CG}(\mathcal{P}^1)$ be a conflict-graph with respect to f_1 over $\mathcal{P}^1$. (see Definition 5.3.4)
   Set $\mathcal{Q}^1$ to be a completion of $\mathcal{Q}^0$ to a MIS of G_1.
5. Form $\hat{E}$, the set of future boundary edges of M_1. $\hat{E}$ is formed by edge contraction of the nodes $\mathcal{P}^1 \setminus \mathcal{Q}^1$. Let $\hat{\mathcal{Q}}^2$ be the set of nodes from $\mathcal{P}^2$ that are outside the γ-protective zones of $\hat{E}$. (see Definition 5.3.5.)
6. Let $G_2 = \mathsf{CG}(\mathcal{Q}^1 \cup \hat{\mathcal{Q}}^2)$ be a conflict-graph with respect to f_1.
   Set $\mathcal{Q}^2$ to be a completion of $\mathcal{Q}^1$ to a MIS of G_2.
7. Set $E_1 = DT(\mathcal{Q}^2, \mathcal{B})$, the edges of a constrained Delaunay triangulation of $\mathcal{Q}^{\in}$.
   Return $M_1 = (\mathcal{Q}^2, E_1, \mathcal{B})$.

Figure 5.6: One level function based coarsening.

rectangle intersects any boundary edge but for e, a truncated version is used. (However, γ can be set to guarantee intersection occurs only with incident boundary edges).

5.3.4 Multi-level schemes

The basic one level scheme takes an initial mesh and a coarsening factor, and generates a coarsened mesh. This simple scheme can be used as a subroutine when generating the entire coarsening sequence in one of the following ways:

1. The coarsening factor is fixed to be $\mathcal{C} = 2$ and mesh M_{i+1} is generated using the one level scheme over M_i. Therefore, at level $i+1$ the node set and spacing function of mesh M_i are used. Figure 5.7 contains a pseudo-code description of the scheme.

   This method generates a node-nested coarsening sequence. However, the mesh quality can repeatedly degrade since the only information used in generating the mesh M_{i+1} is the information embedded in mesh M_i. Surprisingly, we can theoretically prove that repeated degradation does *not* occur when threshold coarsening or quasi-uniform coarsening is employed, and this result is presented in Chapter 6. However, for general coarsening sequences the question is still open. Chapter 8 demonstrates experimentally that repeated degradation does not occur in practice, even over very graded meshes.

2. Always using mesh M_0 as the basis mesh in generating the coarsening sequence meshes. Mesh M_i is generated over mesh M_0 using a coarsening factor $\mathcal{C}_i = 2^i$. Figure 5.8 contains a pseudo-code description of the scheme.

   This scheme guarantees good aspect ratio, since the the initial mesh provides a good point set to sample from and repeated degradation can not occur. However, the sequence is not node nested since the meshes at each level are created independently.

3. A subtler approach, that provides both a node-nested sequence and good aspect ratio guarantees. Similar to the previous approach, M_0 is always used in generating mesh M_i using a coarsening factor $\mathcal{C}_i = 2^i$. As previously, this can be shown to guarantee a good aspect ratio coarsening sequence. However, in order to guarantee that the sequence is node nested as well the sequence is generated in **decreasing** order: mesh M_k, the coarsest mesh, is generated first. The node set of M_k can then forced to be appear in M_{k-1} to generate a node-nested sequence. This is possible since at each level all MIS sets are equally valid. Lemma 5.3.6 shows it is always feasible to force $\mathcal{P}_k \subseteq \mathcal{P}_{k-1}$ in that manner, and later sections show that the resulting scheme produces meshes of good aspect ratio. Figure 5.9 contains a pseudo-code description of the scheme.

Lemma 5.3.6 *Let $\mathcal{P}_i$ be the node set of the coarse mesh M_i generated by the algorithm of Figure 5.6 with input $M_0, f_0, \mathcal{C}_i, \beta, \gamma$. Let $\mathcal{C}_j < \mathcal{C}_i$ be a smaller coarsening factor. The algorithm can be modified to complete $\mathcal{P}_i$ into a coarse mesh M_j, using input parameters $M_0, f_0, \mathcal{C}_j, \beta, \gamma$.*

Proof: $\mathcal{C}_j < \mathcal{C}_i$ implies that $f_j < f_i$. Therefore, any conflict-graph generated for f_i is stricter (has more edges) then the corresponding graph for f_j. By simply adjusting the priorities of the node set $\mathcal{P}_0$ such that the nodes of $\mathcal{P}_i$ are always considered first, it is straightforward to complete $\mathcal{P}_i$ into a MIS of the less constrained conflict-graph.

Furthermore, since the boundary edges of M_j are smaller too, less nodes are thrown out by the γ-protection zones. □

The third multi level scheme, which is the one carrying with it all the theoretical guarantees we desire, uses the following set of coarsening functions to generate the coarsening sequence from the original mesh M_0:

- **for general coarsening:** $\{f_{\mathsf{NN},2^i,\mathcal{P}}\}$, with a spacing constant *beta* that is set depending on the aspect ratio of M_0 (see section 5.5).
- **for quasi-uniform coarsening:** $\{t_{\mathsf{NN},2^i,\mathcal{P}}\}$ and $\beta = 2$.

The sequence thus generated is node-nested. Section 5.7 shows that the sequence meshes are locally similar. Section 5.5 shows that the meshes are also of bounded aspect ratio. Note that in order to show that, it suffices to show that for an arbitrary coarsening factor $\mathcal{C}$ and a coarsening function which is a C coarsening of the original mesh M_0's spacing function, *any* maximal independent set of the conflict graph is of bounded aspect ratio. Section 5.6 shows that the coarsening sequence is size optimal up to a constant factor.

5.4 Properties of coarsening functions

Given an initial, 1-Lipschitz function g, Section 5.3 describes a coarsening strategy based on first coarsening the function g. This strategy creates a new function, called the *coarsening function* of g; see Definition 5.3.2. This section establishes some of the properties of these coarsening functions. First, the coarsening functions are shown to be 1-Lipschitz.

Lemma 5.4.1 (1-Lipschitz) *If g is 1-Lipschitz, then for any $\mathcal{C} > 1$, $f_{g,\mathcal{C},\mathcal{P}}$ and $t_{g,\mathcal{C},\mathcal{P}}$ are 1-Lipschitz.*

Proof: Let $t = t_{g,\mathcal{C},\mathcal{P}}$ and $f = f_{g,\mathcal{C},\mathcal{P}}$.

- t is 1-Lipschitz: Recall $t = \max(g(x), \mathcal{C}\min g)$. Let $x, y \in \Omega$. If $t(x) = g(x)$ and $t(y) = g(y)$ then $|t(x) - t(y)| = |g(x) - g(y)| \le \|x - y\|$, since g is 1-Lipschitz. If $t(x) = \mathcal{C}\min g$ and $t(y) = \mathcal{C}\min g$ then $|t(x) - t(y)| = 0 \le \|x - y\|$. Therefore, assume $t(x) = g(x)$, and $t(y) = \mathcal{C}\min g$. By definition $t(x) > \mathcal{C}\min g$ and $|t(x) - t(y)| = t(x) - t(y) = g(x) - \mathcal{C}\min g \le g(x) - g(y) \le \|x - y\|$, as required.
- f is 1-Lipschitz: Let x, y be two points in Ω. Without loss of generality, $f(x) < f(y)$. Assume further $f(x) = \hat{f}_{\mathcal{C},p_i}(x)$, and $f(y) = \hat{f}_{\mathcal{C},p_j}(y)$. That is, the values of x and y are fixed by two different local coarsening functions. Then,

$$|f(x) - f(y)| = f(y) - f(x) = \hat{f}_{\mathcal{C},p_j}(y) - \hat{f}_{\mathcal{C},p_i}(x) \le \hat{f}_{\mathcal{C},p_i}(y) - \hat{f}_{\mathcal{C},p_i}(x) \le \|x - y\|.$$

  The first inequality uses the fact that at point y, $\hat{f}_{\mathcal{C},p_i}(y) \ge \hat{f}_{\mathcal{C},p_j}(y)$ by definition. The second inequality uses the obvious fact that $\hat{f}_{\mathcal{C},p_i}$ is 1-Lipschitz.

□

Lemma 5.4.2 (linearity and monotonicity) *For any $\mathcal{C}_1, \mathcal{C}_2 \ge 1$*

$$f_{g,\mathcal{C}_1,\mathcal{P}}(x) \le f_{g,\mathcal{C}_2\mathcal{C}_1,\mathcal{P}}(x) \le \mathcal{C}_2 f_{g,\mathcal{C}_1,\mathcal{P}}(x).$$

Procedure: MULTI_LEVEL_COARSEN_1$(M_0, f_0, \beta, \gamma)$

1. Let k be the length of the required coarsening sequence:
2. For $i = 1$ to k

   $(M_i, f_i) =$ ONE_LEVEL_COARSEN$(M_{i-1}, f_{i-1}, 2, \beta, \gamma)$
3. Return $(M_1, ..., M_k)$.

Figure 5.7: Multi level function based coarsening: iterative application.

Procedure: MULTI_LEVEL_COARSEN_2$(M_0, f_0, \beta, \gamma)$

1. Let k be the length of the required coarsening sequence:
2. For $i = 1$ to k

   $M_i =$ ONE_LEVEL_COARSEN$(M_0, f_0, 2^i, \beta, \gamma)$
3. Return $(M_1, ..., M_k)$.

Figure 5.8: Multi level function based coarsening: multi-mesh generation approach.

Procedure: MULTI_LEVEL_COARSEN_3$(M_0, f_0, \beta, \gamma)$

1. Let k be the length of the required coarsening sequence:
2. Let ONE_LEVEL_MODIFIED$(M_0, f_0, \mathcal{C}, \beta, \gamma, \mathcal{P})$ be a modification of ONE_LEVEL_COARSEN that forces $\mathcal{P}$ into the output coarsened mesh. (see Lemma 5.3.6).
3. Let $\mathcal{P}_{k+1} = \emptyset$.
4. For $i = k$ to 1
   - $M_i =$ ONE_LEVEL_MODIFIED$(M_0, f_0, 2^i, \beta, \gamma, \mathcal{P}_{i+1})$
   - Set $\mathcal{P}_i$ to the node set of mesh M_i.
5. Return $(M_1, ..., M_k)$.

Figure 5.9: Multi level function based coarsening: coarse-to-fine approach.

Proof: Since f is a minimum over a collection of functions $\{\hat{f}_{p_i}| \quad p_i \in \mathcal{P}\}$ it suffices to show the inequality for each such function. The first inequality is obvious. The second inequality follows from:

$$\hat{f}_{\mathcal{C}_2\mathcal{C}_1,p_i}(x) = \mathcal{C}_2\mathcal{C}_1 g(p_i) + ||p_i - x|| \leq \mathcal{C}_2(\mathcal{C}_1 g(p_i) + ||p - x||) \leq \mathcal{C}_2 \hat{f}_{\mathcal{C}_1,p_i}(x).$$

□

The following lemma strongly motivates the choice of the coarsening functions: they are the largest possible 1-Lipschitz functions that coarsen the function at the set points by a coarsening factor smaller than or equal to $\mathcal{C}$.

Lemma 5.4.3 (maximality of $f_{\mathcal{C},\mathcal{P}}$) *Let $\mathcal{C} > 1$. Let h be a 1-Lipschitz function over the domain Ω such that for all $p_i \in \mathcal{P}$ $h(p_i) \leq \mathcal{C}g(p_i)$. Then for all $x \in \Omega$, $h(x) \leq f_{g,\mathcal{C},\mathcal{P}}(x)$.*

Proof: $f_{\mathcal{C},\mathcal{P}}$ is a minimum over a collection of functions $\{\hat{f}_{p_i}| \quad p_i \in \mathcal{P}\}$. It suffices to show $h(x) \leq \hat{f}_{\mathcal{C},p_i}(x)$. Recall that $\hat{f}_{\mathcal{C},p_i}(x) = \mathcal{C}g(p_i) + ||p - x||$. Because h is 1-Lipschitz,

$$h(x) \leq h(p_i) + ||p_i - x|| \leq \mathcal{C}g(p_i) + ||p_i - x|| = \hat{f}_{\mathcal{C},p_i}(x).$$

□

The following lemma shows that if the generating point-set (the point set used to derive the initial spacing-function) is fixed, generating the coarsening functions sequence iteratively is equivalent to generating it using geometrically increasing coarsening factors:

Lemma 5.4.4 (iterative construction) *Let $f_i = f_{g,2^i,\mathcal{P}}$. Let $g_{i+1} = f_{g_i,2,\mathcal{P}}$, then $f_i \equiv g_i$.*

Proof: By Lemma 5.4.3, $g_i(x) \leq f_i$. The reverse inequality is shown by induction. Since $g_1 \equiv f_1$, clearly $f_1 \leq g_1$. For the induction case, assume $f_i \leq g_i$. By Lemma 5.4.2 $f_{i+1} \leq 2f_i$ for any point $p \in \mathcal{P}$.

$f_{i+1}(p) \leq 2f_i(p) \leq 2g_i(p)$ implies, by maximality of g_{i+1} with respect to g_i (Lemma 5.4.3 again) that $f_{i+1} \leq g_{i+1}$. □

The threshold coarsening functions, see Section 5.3, are easier to compute than the standard coarsening functions, and should be used whenever possible. The next lemma shows that over quasi-uniform meshes, the threshold coarsening functions are similar to the standard coarsening functions. This implies that all subsequent results about the properties of general coarsening will be valid for threshold coarsening of quasi-uniform meshes.

Lemma 5.4.5 (threshold and general coarsening functions) *Let $M = (\mathcal{P}, E, \mathcal{B})$ be a good aspect-ratio mesh, that is, M has the bounded $\mathcal{L}$-gaps property. Furthermore, suppose M is quasi-uniform, i.e. there exists a constant $\rho \geq 1$ such that for the spacing g of M*

$$\min g \leq g(x) \leq \rho \min g.$$

Then

$$\frac{f_{g,\mathcal{C},\mathcal{P}}}{\rho(\rho + 2\mathcal{L})} \leq t_{g,\mathcal{C},\mathcal{P}} \leq f_{g,\mathcal{C},\mathcal{P}},$$

where $\mathcal{C} > 1$ is some arbitrary coarsening factor.

Proof:

- $t \leq f$: It suffices to show $t \leq f_{g,\mathcal{C},p_i}$. t is 1-Lipschitz, so $t(x) \leq t(p_i) + \|p_i - x\|$. Clearly, $t(p_i) \leq f_{\mathcal{C},p_i}p_i = \mathcal{C}g(p_i)$. Hence, $t(x) \leq \mathcal{C}g(p_i) + \|p_i - x\| = f_{g,\mathcal{C},p_i}(x)$.

- $\rho(\rho + 2\mathcal{L})t \geq f$: M has the bounded $\mathcal{L}$-gaps property, so a ball of radius $\mathcal{L}g(x)$ through x must contain some node $p_i \in \mathcal{P}$. Hence:

$$\begin{aligned} f(x) \quad & \leq \mathcal{C}g(p_i) + \|x - p_i\| \\ & \leq \mathcal{C}\rho g(x) + 2\mathcal{L}g(x) \\ & \leq \mathcal{C}(\rho^2 + 2\mathcal{L}\rho) \min g \\ & \leq \rho(\rho + 2\mathcal{L})\mathcal{C} \min g \\ & \leq \rho(\rho + 2\mathcal{L})t(x) \end{aligned}$$

□

5.5 Aspect-ratio bounds for one-level mesh coarsening

This section concentrates on proving that the one-level coarsening algorithm produces a bounded aspect ratio mesh. For the specification of the algorithm see Figure 5.6, Section 5.3.

The aspect ratio bounds will be shown for two cases: general unstructured coarsening, and coarsening of quasi-uniform meshes. The latter is much simpler to prove and also allows for deriving better bounds.

The sections below concentrate on bounding the gaps in the meshes produced by algorithm ONE_LEVEL_COARSEN. Refering back to Chapter 2, a constrained Delaunay triangulation of a point set is of bounded aspect ratio if (1) the point set is well-spaced, and (2) points are not to close, relative to the spacing function, to a boundary segment that they are not incident to.

To show that the point set is well-spaced, points should be spaced by a spacing function and the gaps bounded. The nodes of the mesh generated by algorithm ONE_LEVEL_COARSEN are spaced according to a spacing function by their construction. Nodes are not too close to boundary segments because of the boundary segments protective zones employed by the algorithm. The main technical challenge that remains is that of bounding the gaps relative to the spacing function.

5.5.1 Aspect ratio bound for one iteration coarsening of general meshes

Algorithm ONE_LEVEL_COARSEN takes as input a good aspect ratio mesh $M_0 = (\mathcal{P}_0, E_0, \mathcal{B})$, a desired coarsening factor $\mathcal{C}$, and generates a $\mathcal{C}$-coarsening $M_1 = (\mathcal{P}_1, E_1, B)$. $\mathcal{M}_0$ is a good aspect ratio mesh, so by Theorem 2.4.1 its point set is well-spaced with respect to some spacing function f_0. Furthermore, it has the bounded gaps property with constant parameter $\mathcal{L}_0$. To show M_1, the output mesh, is a good aspect ratio mesh, it suffices – by Theorem 2.5.1 – to show that M_1 has the bounded gaps property with constant gap parameter $\mathcal{L}_1$ that depends on $\mathcal{L}_0$ only. Recall that the nodes of M_1 are generated by first defining the conflict graph over the set of nodes $\mathcal{P}_0$ with respect to the coarsening function f_1, and then picking a maximal independent set of the conflict graph. The proofs proceed by bounding the gaps for every possible choice of a maximal independent set.

Assumpsions: M_1 is constructed by coarsening the spacing function f_0. The resulting spacing function f_1 is a $\mathcal{C}$ coarsening of the function f_0 over the point set $\mathcal{P}_0$. In this section, the only assumption made is that M_1 is constructed over M_0 using a spacing function $f_1 \geq f_0$. The coarsening

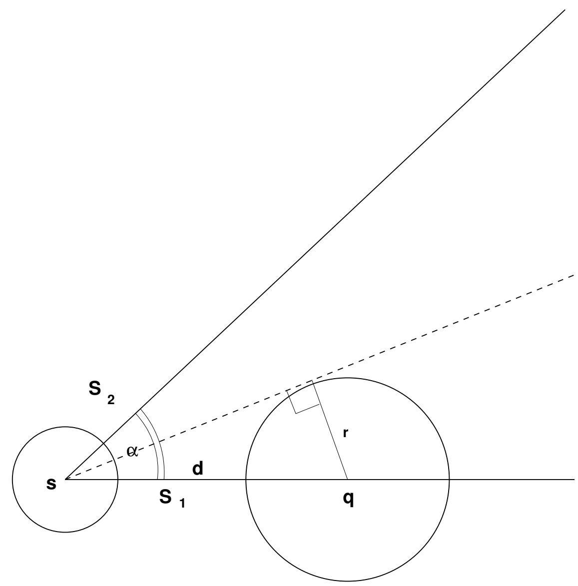


Figure 5.10: If q's ball is to remain below the bisector, then $f(q)/\beta$ must be smaller than $d/\sqrt{2}$.

parameter $\mathcal{C}$ thus plays no role in the proofs. Furthermore, it is assumed that $f_1 \leq \mathsf{lfs}_{\mathcal{B}}$, i.e. that coarsening is subjected to the limitation posed by the local feature size of the boundary description. Furthermore, the boundary description $\mathcal{B}$ is assumed to contain no angle smaller than 60°. This is an often used restriction, see for example [64, 9].

Coarsening parameters: The other parameters used in the coarsening algorithm are the spacing constant β and the edge protective zone constant γ. Note that whereas in a multi-level scheme $\mathcal{C}$, the coarsening factor, changes from level to level, β and γ are fixed during the coarsening sequence generation. The goal of the proofs in this section is show that for a large enough setting of β and a small enough setting of γ, both depending only on the gaps of the mesh M_0, the mesh M_1 has the bounded gaps property.

The following lemma establishes the fact that nodes on different boundary elements do not share conflict graph edges, so different segments can be coarsened independently. We use the fact that the node-set $\mathcal{P}_1$ is generated from $\mathcal{P}_0$ by taking a maximal independent set of the conflict graph formed over the set $\mathcal{P}_0$. Also, recall that the f-ball of a point p is a ball of radius $f(p)/\beta$ centered at p.

Lemma 5.5.1 (Boundary independence) *Let $M_0 = (\mathcal{P}_0, E_0, \mathcal{B})$. Let f_1 be a spacing function such that $f_1 \leq \mathsf{lfs}_{\mathcal{B}}$. Let $\alpha > 60°$ be the minimal angle between two segments of $\mathcal{B}$. Let the spacing parameter $\beta \geq \frac{1}{2\sin\frac{\alpha}{2} - 1}$. The following are true during the coarsening algorithm:*

1. *Let $S_1, S_2 \in \mathcal{B}$ be two disjoint elements of the boundary. S_i can be either a boundary segment or a boundary node. Two f_1-balls placed on S_1 and S_2 can not intersect.*

2. *Let $S_1, S_2 \in \mathcal{B}$ be two incident boundary segments, $s = S_1 \cap S_2 \in \mathcal{P}_0$ their common boundary node. Let $p \in S_1 \cap \mathcal{P}_0$ and $q \in S_2 \cap \mathcal{P}_0$. If the f_1-balls of p and q intersect then they also intersect the f_1-ball of s. As s must be in the MIS by the algorithm specification, this implies*

that removing the conflict graph edges between p and q does not change the outcome of taking the MIS.

Proof:

1. Let $p \in \mathcal{P}_0$ be a node on S_1. Let $q \in \mathcal{P}_0$ be a node on S_2. Since they are on disjoint boundary elements, $f_1(p) \leq \mathsf{lfs}_{\mathcal{B}}(p) \leq \|p - q\|$ and similarly for q. Clearly, if $\beta \geq 2$ the $\frac{f_1}{\beta}$-balls do not intersect.

2. We assume that p's f_1-ball does not intersect that of s, i.e., letting $\ell = \|s - q\|$, that $f_1(s) + f_1(p) \leq \beta\ell$, and show that p's f_1-ball can not intersect the f_1-ball of a similar point q on S_2.

   To show p's ball does not intersect any ball of a similar node on S_2 it suffices to show that p's f_1-ball lies below the bisector between S_1 and S_2. This happens when the radius r of the ball is $r = f_1(p)/\beta \leq \ell \sin(\alpha/2)$. Or, equivalently, if $f_1(p) \leq \beta\ell\sin(\alpha/2)$

   f_1 is 1-Lipschitz, hence for some $-1 \leq \gamma \leq 1$, $f_1(p) = f_1(s) + \gamma\ell$. The previous facts imply $2f_1(p) - \gamma\ell < \beta\ell$, or $f_1(p) \leq \ell(\beta + \gamma)/2 \leq \ell(\beta + 1)/2$. It suffices to require then that $\ell\frac{\beta+1}{2} \leq \beta\ell\sin\frac{\alpha}{2}$ or that $\beta \geq (2\sin\frac{\alpha}{2} - 1)^{-1}$. This also clarifies the need to restrict $\alpha > 60°$.

□

Theorem 5.5.2 (Bounded boundary gaps) *Let mesh $M_0 = (\mathcal{P}_0, E_0, \mathcal{B})$ be a bounded aspect ratio mesh, i.e., it has the bounded $\mathcal{L}_0$-gaps property. Let $M_1 = (\mathcal{P}_1, E_1, \mathcal{B})$ be a coarsening of M_0 by algorithm* ONE_LEVEL_COARSEN *using coarsening function f_1, and a spacing constant $\beta \geq 1$. The boundary gaps of M_1 can be bounded using the constant $\hat{\mathcal{L}}_1$:*

$$\hat{\mathcal{L}}_1 = \mathcal{L}_0 + \frac{4(1+\mathcal{L}_0)}{\beta - 1}\left[1 + \frac{1}{\beta - 1}\right]$$

Proof: Let $S \in \mathcal{B}$ be some boundary segment. By Lemma 5.5.1 only nodes $p \in \mathcal{P}_0 \cap S$ on the same segment can influence the size of the gaps on S. Furthermore, by the same lemma and the priority given to the boundary segments end-points by the algorithm, the end-points of S must belong to $\mathcal{P}_1$. Let $x \in S$ be an arbitrary point on the segment. Without loss of generality let $f_1(x) = 1$. Let ℓ be the length of $\mathcal{I}$, a sub-segment of S starting at x. The proof proceeds by showing how to set $\hat{\mathcal{L}}_1$ such that if $\ell \geq \hat{\mathcal{L}}_1$ then $\mathcal{I} \cap \mathcal{P}_1$ is not empty.

Split $\mathcal{I}$ into three smaller intervals $\mathcal{I}_1$, $\mathcal{I}_2$ and $\mathcal{I}_3$, as in Figure 5.11. The scheme of the proof is to argue that the sizes ℓ_1, ℓ_2 and ℓ_3 of the three intervals can be set such that:

- $\mathcal{I}_2$ must contain a node of $\mathcal{P}_0$,
- a node of $\mathcal{P}_0 \cap \mathcal{I}_2$ can not be eliminated from the MIS of the conflict graph by any other node of $\mathcal{P}_0$ which is outside $\mathcal{I}$ since they cannot share a conflict graph edge; the conclusion being that $\mathcal{I} \cap \mathcal{P}_1$ is not empty.

By the Lipschitz condition, $f_1(w) \leq 1 + \|x - w\|$ for any point $w \in \Omega$. Equality – $f_1(w) = 1 + \|x - w\|$ – can be assumed since once non-intersection is proven for larger balls, then the actual balls must be disjoint as well. Furthermore, looking at Figure 5.11, it suffices to show that the balls $\mathsf{B}(y, f_1(y)/\beta)$ and $\mathsf{B}(z, f_1(z)/\beta)$ can not intersect a sphere centered outside $\mathcal{I}$. Other balls, placed on $\mathcal{I}_2$ between y and z, must be also disjoint from outside balls since their radius grows slowly with the distance.

Setting ℓ_1: claim: if $\ell_1 \geq \frac{2}{\beta-1}$ then $\mathsf{B}(y, f_1(y)/\beta)$ is disjoint from any ball placed left of x. Let u be some point left of x, then $\|u-y\| \geq \ell_1$.

(1)	$f_1(u) = 1 + \|y-u\| - \ell_1$	Lipschitz condition on f_1
(2)	$f_1(u) + f_1(y) \leq 2 + \|y-u\|$	follows from (1)
(3)	$2 + \|y-u\| \leq \beta\|y-u\|$	since $\|y-u\| \geq \ell_1 \geq \frac{2}{\beta-1}$
(4)	$f_1(u) + f_1(y) \leq \beta\|y-u\|$	follows from (2) and (3)

Hence, the balls around y and u do not intersect. For the rest of this proof, set:

$$\boxed{\ell_1 = \frac{2}{\beta-1}}$$

Setting ℓ_2: claim: if $\ell_2 \geq \mathcal{L}_0(1+\frac{2}{\beta-1})$ then $\mathcal{I}_2 \cap \mathcal{P}_0$ is not empty.
Mesh M_0 has boundary gaps bounded by $\mathcal{L}_0$, hence an interval of length $\mathcal{L}_0 f_0(y)$ must contain a point of $\mathcal{P}_0$. The interval of length $\mathcal{L}_0 f_1(y)$ is even larger and hence also contains a point of $\mathcal{P}_0$. Using $f_1(y) \leq 1 + \ell_1 = 1 + \frac{2}{\beta-1}$ leads to the result. For the rest of this proof, set

$$\boxed{\ell_2 = \mathcal{L}_0(1+\frac{2}{\beta-1})}$$

Setting ℓ_3: claim: if $\ell_3 \geq \frac{2+2\mathcal{L}_0}{\beta-1}\left(1+\frac{2}{\beta-1}\right)$ then $\mathsf{B}(z, f_1(z)/\beta)$ is disjoint from any ball placed right of $\mathcal{I}$. Another way of writing the value of ℓ_3 is:

$$2(1+\mathcal{L}_0)\left(1+\frac{2}{\beta-1}\right) + \ell_3 \leq \beta\ell_3$$

(1)	$f(z) = 1 + \ell_1 + \ell_2 = (1+\mathcal{L}_0)\left(1+\frac{2}{\beta-1}\right)$	previous settings of ℓ_1 and ℓ_2
(2)	$f(v) = f(z) + \|v-z\|$	
(3)	$f(z) + f(v) \leq 2f(z) + \|v-z\| = 2(1+\mathcal{L}_0)\left(1+\frac{2}{\beta-1}\right) + \|v-z\|$	
(4)	$f(z) + f(v) \leq \beta\|v-z\|$	by the setting of ℓ_3, and the fact $\|v-z\| \geq \ell_3$.

Summing the interval lengths, the following setting of $\hat{\mathcal{L}}_1$ suffices to guarantee the interval is not empty:

$$\boxed{\hat{\mathcal{L}}_1 = \mathcal{L}_0 + \frac{4(1+\mathcal{L}_0)}{\beta-1}\left[1+\frac{1}{\beta-1}\right]}$$

□

Remarks: The proof for the boundary coarsening case might be lengthy, but is very straightforward. Shorter proofs are possible, at the cost of deriving a weaker bound on $\hat{\mathcal{L}}_1$ and using stronger restrictions on β. It is interesting to notice that the proof above poses no restrictions on β and that $\hat{\mathcal{L}}_1$ can be set as close to $\mathcal{L}_0$ as is wished for, by increasing β. Nonetheless, it is always the case that $\hat{\mathcal{L}}_1 > \mathcal{L}_0$. This implies that the coarsening degrades the aspect ratio. Note however that this is a one

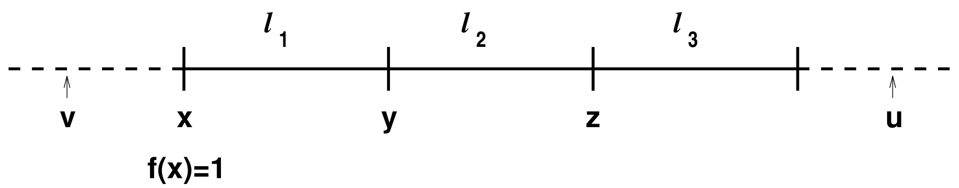


Figure 5.11: Bounded gaps of the coarsened boundary — Theorem 5.5.2.

time degradation rather than repeated degradation. The main goal of coarsening is to use strategies that do not surrender to repeated degradation of the aspect ratio.

In two dimension, the bounded gaps proof is more complex. This is also be expressed in strong restrictions on the spacing constant β. In two dimension the gaps take the form of balls (or squares) and are much harder to work with than the simple segment-like gapes in the one dimensional boundary case.

Theorem 5.5.3 (Interior gaps of 2D coarsening) *Let mesh $M_0 = (\mathcal{P}_0, E_0, \mathcal{B})$ be a bounded aspect ratio mesh, i.e., it has the bounded $\mathcal{L}_0$-gaps property. Let $f_1 \geq f_0$ be the coarsening function. Let $M_1 = (\mathcal{P}_1, E_1, \mathcal{B})$ be the coarser mesh, obtained using spacing parameter $\beta = 12\sqrt{\mathcal{L}_0}$, and boundary protection zone parameter $\gamma = \frac{1}{\hat{\mathcal{L}}_1(8+\mathcal{L}_0)}$. Then, mesh M_1 has the bounded gaps property with parameter $\mathcal{L}_1$, where*

$$\mathcal{L}_1 = \max\{4, 12\mathcal{L}_0, (4\hat{\mathcal{L}}_1^2 + 1)(3\mathcal{L}_0 + 2)\}.$$

Proof:

Let $u \in \Omega$ be an arbitrary point in the domain, B a ball through u representing an interior gap at u. Consider Figure 5.12. Without loss of generality, let $f_1(u) = 1$, $u = (0,0)$, and let the center of B lie on the y-axis. B is an interior gap of M_1, so assume the radial (u, c) from u to the center of B is contained in the domain.

Let R be the radius of B. This proof shows that if R is set to $\mathcal{L}_1$, which is a large constant, B must contain a point of $\mathcal{P}_1$ visible to u and thus M_1 has the bounded $\mathcal{L}_1$-gaps property.

Let B_0 be a smaller ball nested in B of radius $3\mathcal{L}_0$ centered at $(0, 2 + 3\mathcal{L}_0)$. Let $v = (0, 2)$, a point on the circumference of B_0. $f_1(v) \leq f_1(u) + 2 = 3$. Since the radial from v is in the domain, and the radius of B_0 is larger or equal to $\mathcal{L}_0 f_0(v)$, B_0 is a bounded gap of M_0 and contains a node $q \in \mathcal{P}_0$ visible to v. By Lemma 5.5.4, either q is visible to u as well, or the ball B contains a node p visible to u which is a segment end-point and hence $p \in \mathcal{P}_1$, the gap B is not empty and the proof is done. Therefore, assume q is visible to u. The proof proceeds as follows:

1. If $R \geq \max(12\mathcal{L}_0, 4)$ then by Lemma 5.5.5 the node q can not be eliminated from $\mathcal{P}_1$ by any node outside B.

2. If $R \geq (4\hat{\mathcal{L}}_1^2 + 1)(3\mathcal{L}_0 + 2)$ then no boundary segment protection can eliminate q from $\mathcal{P}_1$, by Lemma 5.5.8.

Hence, $q \in \mathcal{P}_1$ or it is the case it is eliminated by some point $p \in B \cap \mathcal{P}_1$ visible to q. Then again by Lemma 5.5.4 B must contain a point of $\mathcal{P}_1$ visible to u, and M_1 possesses the bounded gaps property. □

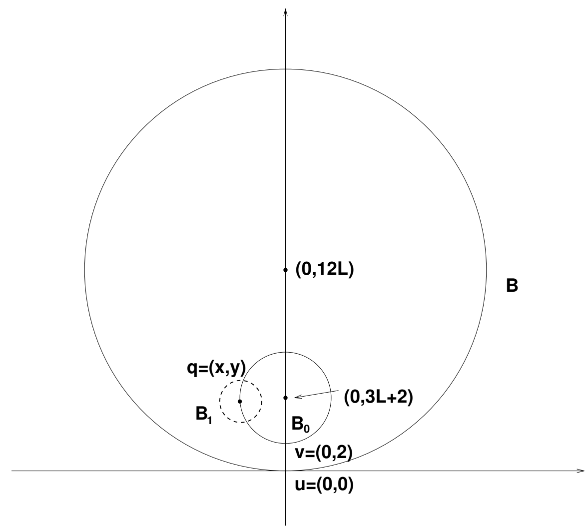


Figure 5.12: After one level coarsening, a circle of radius $12\mathcal{L}_0 f_1(u)$ must contain a node of $\mathcal{P}_1$

Lemma 5.5.4 (Visibility) *Let B be a ball through some point $u \in \Omega$. Let v be some point visible to u, and w is a point visible to v. Assume further that $v, w \in B$. If w is not visible to u then some end-point of a boundary segment is in B and visible to u.*

Proof: Look at the triangle with end-points (u, v, w). This triangle is contained in B by convexity. Let S be the segment intersecting triangle edge (u, w) closest to u. Since w is not visible to u, such a segment exists. No segment can intersect both the edge (u, v) and the edge (v, w). Therefore, one of the end-points of S, call it p, must be in the triangle. Moreover, no other segment intersecting (u, w) can obscure p from u.

Note however that segments totally contained in the triangle can obscure p from u. Let q be the nearest point to u from the set of segment end-points in B. q exists and must be visible to u. □

Lemma 5.5.5 *If $R \geq \max(4, 12\mathcal{L}_0)$, and $\beta \geq \max(3, 12\sqrt{\mathcal{L}_0})$, then no point p placed outside B can eliminate q from $\mathcal{P}_1$.*

Proof: Refer back to Theorem 5.5.3 for the setting. Without loss of generality, let $q = (x, y)$ be on the boundary of B_0. (all the bounds derived in that case still hold if q is internal to B_0).
The following properties of q are used:

I) By Lemma 5.5.6, the distance from q to the boundary of B is larger than $y/2$.

II) By Lemma 5.5.7, if $\mathcal{L}_0 \geq 4$ and $\beta = 12\sqrt{\mathcal{L}_0}$, then $f(q)/\beta \leq y/6$. Note that we may increase $\mathcal{L}_0$ since if a mesh has the bounded $\mathcal{L}$-gaps property, it also has the bounded gaps property for a larger $\mathcal{L}$.

Let B_1 be the ball with center q and radius $f_1(q)/\beta$ (see the dashed ball in Figure 5.12). Let $p \in \mathcal{P}_1$ be some point placed outside B. The goal is to show its ball can not intersect B_1. Property

I implies that $\ell = \|p - q\| \geq y/2$. Thus,

$$\begin{array}{rcll} f_1(p) & \leq & \ell + f_1(q) & \text{(because } f \text{ is Lipschitz)} \\ f_1(q) + f_1(p) & \geq & \beta\ell & \text{(because they share a conflict graph edge)} \\ 2f_1(q) + \ell & \geq & \beta\ell & \text{(follows from the two previous lines)} \\ 2f_1(q) & \geq & (\beta - 1)\ell \geq (\beta - 1)y/2 & \text{(by property I)} \\ \beta y/3 & \geq & 2f_1(q) \geq (\beta - 1)y/2 & \text{(by property II)} \\ \beta/3 & \geq & (\beta - 1)/2 & \end{array}$$

This is a contradiction when $\beta > 3$ and hence q and p have disjoint balls. □

Lemma 5.5.6 *If $\mathcal{L}_0 > 2/3$ then the distance from any point (x, y) on the boundary of B_0 to the boundary of B is greater than $y/2$.*

Proof: First notice that $y > 0$. The distance from (x, y) to the boundary of B is equal to $R - \sqrt{x^2 + (y - R)^2}$, where recall that $R = 12\mathcal{L}_0$. Because (x, y) is on the boundary of B_0, we have $x^2 + (y - r - 2)^2 = r^2$, implying $x^2 + y^2 = 2y(2 + r) - 4 - 4r = y(4 + 6\mathcal{L}_0) - 4 - 12\mathcal{L}_0$, for $r = 3\mathcal{L}_0$. Thus,

$$\begin{array}{rcl} x^2 + (y - R)^2 & = & x^2 + y^2 - 2yR + R^2 \\ & = & y(4 - 18\mathcal{L}_0) - 4 - 12\mathcal{L}_0 + 144{\mathcal{L}_0}^2 \\ & = & (12\mathcal{L}_0 - y/2)^2 - 144{\mathcal{L}_0}^2 + 12y\mathcal{L}_0 - y^2/4 + 4y - 18y\mathcal{L}_0 - 4 - 12\mathcal{L}_0 + 144{\mathcal{L}_0}^2 \\ & = & (12\mathcal{L}_0 - y/2)^2 - y^2/4 + y(4 - 6\mathcal{L}_0) - 4 - 12\mathcal{L}_0 \\ & < & (R - y/2)^2. \end{array}$$

The last inequality is true because $\mathcal{L}_0 > 2/3$, $y > 0$, and $R = 12\mathcal{L}_0$. Because $y/2 < R$, we have $R - \sqrt{x^2 + (y - R)^2} > y/2$, completing the proof. □

Lemma 5.5.7 *If $\mathcal{L}_0 > 4$ and $\beta = 12\sqrt{\mathcal{L}_0}$, then*

$$\frac{f_1(q)}{\beta} \leq \frac{1 + \|u - q\|}{\beta} \leq \frac{y}{6}.$$

Proof:

As shown in the proof of Lemma 5.5.6 $\|u - q\|^2 = x^2 + y^2 = y(4 + 6\mathcal{L}_0) - 4 - 12\mathcal{L}_0$. Thus, $f_1(q) \leq 1 + \|p - q\| \leq 1 + \sqrt{y(4 + 6\mathcal{L}_0) - 4 - 12\mathcal{L}_0}$. The goal is to show $1 + \sqrt{y(4 + 6\mathcal{L}_0) - 4 - 12\mathcal{L}_0} \leq \beta y/6$, or equivalently, $\sqrt{y(4 + 6\mathcal{L}_0) - 4 - 12\mathcal{L}_0} \leq \beta y/6 - 1$

Squaring both side, we thus need to show $y(4 + 6\mathcal{L}_0) - 4 - 12\mathcal{L}_0 < \beta^2 y^2/36 + 1 - \beta y/3$.

This is true if $\beta^2 y^2/36 + 5 + 12\mathcal{L}_0 - \beta y/3 - y(4 + 6\mathcal{L}_0) > 0$. It suffices to show that $\beta^2 y^2/36 - \beta y/3 - y(4 + 6\mathcal{L}_0) > 0$.

Because $y > 0$, dividing by y, we need to show that $\beta^2 y/36 - \beta/3 - (4 + 6\mathcal{L}_0) > 0$. Since $y > 2$ this is true if $\beta^2/18 - \beta/3 - (4 + 6\mathcal{L}_0) > 0$. Notice that the conditions $\mathcal{L}_0 > 4$ and $\beta = 12\sqrt{\mathcal{L}_0}$ guarantees this inequality. □

Lemma 5.5.8 . *If $R \geq (4\hat{\mathcal{L}}_1^2 + 1)(3\mathcal{L}_0 + 2)$ and $\gamma \leq \frac{1}{\hat{\mathcal{L}}_1(8 + 2\mathcal{L}_0)}$ then q can not be eliminated from $\mathcal{P}_1$ by the protection zone around any boundary edge $e \in E_1$ of M_1.*

Proof: Note that $e \in E_1$ is a boundary edge, fully contained in some boundary segment S. The proof is split into two parts: first, R is set so that no segment can get too close to the ball B_0, and second, γ is set so that the protection zone around the farther intervals do not eliminate q.

In either case, it can be assumed that the edge end-points are not inside B. The reasoning is as follows: assume that e is some edge such that q is within its protective zone. Note that since e must be visible to q, and no other segment's end-point can intrude on the protective zone by local feature size considerations, the two nodes of e are visible to q. If any of these nodes are in B, then by Lemma 5.5.4, they are visible to u as well and the proof that B contains a node visible to u is completed. Hence, from all relevant edges, i.e. those whose protective zone can contain q, it is necessary to consider only edges that have no end-nodes in B.

1) **Segments do not get too close to B_0:** Look at a ring G of width 1 around B_0. A point $z \in G$ can be written as:

$$z = (x, y) \qquad x^2 + (y - (3\mathcal{L}_0 + 2))^2 = (3\mathcal{L}_0 + \epsilon)^2 \qquad 0 \leq \epsilon \leq 1$$

or:

$$x^2 + y^2 = (3\mathcal{L}_0 + \epsilon)^2 - (3\mathcal{L}_0 + 2)^2 + 2y(2\mathcal{L}_0 + 2)$$

Let $\mathcal{I}$ be a segment of $\mathcal{M}_1$ through z. If any of $\mathcal{I}$'s end points is inside B then B is not empty, and the proof is done. Otherwise, $\mathcal{I}$'s end points are outside B. The shortest segment with no end points in B through z is the chord through z. Therefore, the proof scheme will show that the gap at z must be even shorter than a chord. Let s be half the length of a chord through z, then $s = \sqrt{R^2 - (x^2 + (y - R)^2)} = \sqrt{2yR - (x^2 + y^2)}$. The length of an empty 1D gap at z, by Theorem 5.5.2, must be smaller than $\hat{\mathcal{L}}_1 f_1(z) \leq \hat{\mathcal{L}}_1(1 + \sqrt{x^2 + y^2})$.

We now show that R can be set so that the maximal gap at z is smaller than half the chord length at z. This implies that $\mathcal{I}$ has an endpoint in B - a contradiction.

(1)	$\hat{\mathcal{L}}_1(1 + \sqrt{x^2 + y^2}) < \sqrt{2yR - (x^2 + y^2)}$	the goal inequality to show
(2)	$2\hat{\mathcal{L}}_1\sqrt{x^2 + y^2}) < \sqrt{2yR - (x^2 + y^2)}$	stronger inequality, since $y \geq 1$
(3)	$4\hat{\mathcal{L}}_1^2(x^2 + y^2) < 2yR - (x^2 + y^2)$	squaring (2)
(4)	$(4\hat{\mathcal{L}}_1^2 + 1)\left(3\mathcal{L}_0 + 2 - \frac{(3\mathcal{L}_0+2)^2-(3\mathcal{L}_0+\epsilon)^2)}{2y}\right) < R$	using the value for $x^2 + y^2$
(5)	$(4\hat{\mathcal{L}}_1^2 + 1)\,(3\mathcal{L}_0 + 2) < R$	stronger inequality, since the second term on the left hand side is always negative

Therefore, it suffices to set:

$$\boxed{R > (4\hat{\mathcal{L}}_1^2 + 1)\,(3\mathcal{L}_0 + 2)}$$

2) **Far boundary edges can not eliminate q:** Let d be the distance between q and some boundary edge $\mathcal{I}$. The edge $\mathcal{I}$ is considered "far" if $d > 1$. In the previous part of this proof, R was set so that no boundary edge is close to q.

Since $q \in B_0$, $f_1(q) \leq 3 + 6\mathcal{L}_0$. Let $z \in \mathcal{I}$ be the point closet to q. $f_1(z) \leq 3 + 6\mathcal{L}_0 + d$. Therefore, the length of $\mathcal{I}$ is bounded (see Theorem 5.5.2):

$$|\mathcal{I}| \leq 2\hat{\mathcal{L}}_1(3 + 6\mathcal{L}_0 + d)$$

The protective zone around $\mathcal{I}$ is a rectangle projecting $\gamma|\mathcal{I}|$ around $\mathcal{I}$. For $\mathcal{I}$ not to eliminate q, γ should be set so that $\gamma|\mathcal{I}| < d$, or

$$\gamma 2\hat{\mathcal{L}}_1(3 + 6\mathcal{L}_0 + d) \leq d$$

Rearranging, this reduces to:

$$\frac{\gamma \hat{\mathcal{L}}_1(6 + 2\mathcal{L}_0)}{1 - 2\hat{\mathcal{L}}_1\gamma} \leq d$$

or, since $d > 1$, it suffices to require:

$$\frac{\gamma \hat{\mathcal{L}}_1(6 + 2\mathcal{L}_0)}{1 - 2\hat{\mathcal{L}}_1\gamma} \leq 1$$

or

$$\boxed{\gamma \leq \frac{1}{\hat{\mathcal{L}}_1(8+2\mathcal{L}_0)}}$$

□

5.5.2 Aspect ratio bound for coarsening of quasi-uniform meshes

The gaps derived for the general case apply to quasi-uniform meshes as well. Nonetheless, stronger results are possible for this special case:

- the spacing factor β can be set independently of the gap parameter $\mathcal{L}_0$ of the original mesh.
- the coarsened mesh can be shown to have the same bounded gaps parameter – i.e. there is no repeated degradation in the mesh quality. This implies that the simpler scheme MULTI_LEVEL_COARSEN_1, rather than MULTI_LEVEL_COARSEN_3, can be used.

Not only are the results stronger, but the proofs and the algorithms for this case are also simpler. It is interesting to point out that in contrast to the general coarsening theorems, the proofs actually use the fact that the coarsening factor is at least 2. In the general coarsening case one can not deduce that the coarsening function value will change much when a large coarsening value is used, whereas in the quasi-uniform case the larger the coarsening factor is the greater the change.

The following proves the result for the simple case where boundary effects are ignored:

Theorem 5.5.9 (Bounded gaps of quasi-uniform coarsening) *Let mesh $M_0 = (\mathcal{P}_0, E_0, \mathcal{B})$ be a quasi-uniform, bounded aspect ratio mesh with bounded gaps parameter $\mathcal{L}_0$. Let $\mathcal{C} > 2$ be the coarsening factor. Let $M_1 = (\mathcal{P}_1, E_1, \mathcal{B})$ be the coarser mesh returned by algorithm* ONE_LEVEL_COARSEN *with spacing parameter $\beta = 2$. Then, the mesh M_1 has the bounded gaps property with parameter $\mathcal{L}_0$.*

Proof: Without loss of generality, it can be assumed that f_0 is constant, and that $f_1 = \mathcal{C} f_0$. Since the original mesh is quasi-uniform, a few applications of coarsening using the threshold coarsening function leads to a constant function.

Fix $u = (0, 0) \in \Omega$. Assume $f_1(u) = 1$. Let B be a ball of radius $\mathcal{L}_0 f_1$. Let B_0 be a smaller ball centered inside B, of radius $\mathcal{L}_0 f_0$. There exists a point $q \in \mathcal{P}_0 \cap B_0$ by the bounded gaps of mesh M_0. The distance from q to the boundary of B is at least $\mathcal{L}_0(f_1 - f_0)$. If $f_1/\beta \leq \mathcal{L}_0(f_1 - f_0)/2$ then the f_1/β ball around q can intersect no similar ball centered outside of B. Since $\beta = 2$, this is true if $f_1 \leq \mathcal{L}_0(f_1 - f_0)$ or if $\mathcal{L}_0/(\mathcal{L}_0 - 1) \leq \mathcal{C}$. This holds whenever $\mathcal{C}, \mathcal{L}_0 \geq 2$.

□

5.6 Size optimality of one-level coarsening

The size optimality of mesh M_1, the output mesh of algorithm ONE_LEVEL_COARSEN, is a result of the optimality of the coarsening function used by the algorithm. The number of elements in a bounded aspect ratio mesh is directly related to its spacing function. We start by showing that the integral $\frac{1}{\mathsf{NN}^2}$ of a bounded aspect ratio mesh over one of its triangles is a constant that depends only on the triangle's aspect ratio. We then use this fact, together with the optimality of the coarsening functions, to obtain a proof of the size optimality of our meshes.

Lemma 5.6.1 *Let M be a mesh with smallest angle θ. Let NN be the nearest neighbor function of the mesh. Let T be a triangle of M, then there exist two positive constants C_1 and C_2 such that*

$$\frac{\sin^2\theta}{2{C_2}^2} \leq \int_T \frac{dA}{\mathsf{NN}_M^2} \leq \frac{1}{2C_1^2\sin\theta}.$$

Proof: Let $\mathsf{el}(T)$ be the longest side of T, and let α, β, and γ be its angles. The area of T is

$$A(T) = 0.5\mathsf{el}^2(T)\sin\alpha\sin\beta/\sin\gamma$$

Using the bound θ on the smallest angle,

$$0.5\mathsf{el}^2(T)\sin^2\theta \leq A(T) \leq 0.5\mathsf{el}^2(T)/\sin\theta$$

By Theorem 2.2.9

$$\frac{A(T)}{{C_2}^2 el^2(T)} \leq \int_T \frac{dA}{\mathsf{NN}_M^2} \leq \frac{A(T)}{{C_1}^2 el^2(T)},$$

or, using the bounds on $A(T)$,

$$\frac{\sin^2\theta}{2{C_2}^2} \leq \int_T \frac{dA}{\mathsf{NN}_M^2} \leq \frac{1}{2C_1^2\sin\theta}.$$

□

Corollary 5.6.2 *The number of elements of mesh M with smallest angle θ is:*

$$a_1(\theta)\int_\Omega \frac{dA}{\mathsf{NN}^2} \leq |M| \leq a_2(\theta)\int_\Omega \frac{dA}{\mathsf{NN}^2}$$

where $a_1(\theta)$ and $a_2(\theta)$ are constants depending only on θ.

Theorem 5.6.3 (Size optimality of one-level coarsening) *Consider the following three meshes:*

- *M_0, a mesh with smallest angle θ.*
- *M_1, the coarsening of M_0 generated by algorithm* ONE_LEVEL_COARSEN *with coarsening factor $\mathcal{C}$ and a coarsening constant β.*
- *M_1', a mesh with smallest angle θ', whose elements are at most a factor $\mathcal{C}$ greater than M_0, i.e. $\mathsf{el}_{M_1'} \leq \mathcal{C}\mathsf{el}_{M_0}$.*

There exists a constant b, depending on θ and θ' only, such that $|M_1| \leq b|M_1'|$.

Proof: Since $\mathsf{el}_{M_1'} \leq \mathcal{C}\mathsf{el}_{M_0}$, Theorem 2.2.9 implies:

$$\mathsf{NN}_{M_1'} \leq C_2(\theta')\mathsf{el}_{M_1'} \leq \mathcal{C}C_2(\theta')\mathsf{el}_{M_0} \leq \frac{\mathcal{C}C_2(\theta')}{C_1(\theta)}\mathsf{NN}_{M_0},$$

or,

$$\mathsf{NN}_{M_1'} \leq a\mathcal{C}\mathsf{NN}_{M_0}$$

where $a = \max\left(1, \frac{C_2(\theta')}{C_1(\theta)}\right)$. Since $\mathsf{NN}_{M_1'}$ is 1-Lipschitz, by Lemma 5.4.3:

$$\mathsf{NN}_{M_1'} \leq f_{\mathsf{NN}_{M_0}, a\mathcal{C}, \mathcal{P}_0}$$

By Lemma 5.4.2, using the fact $a \geq 1$:

$$\mathsf{NN}_{M_1'} \leq a f_{\mathsf{NN}_{M_0}, \mathcal{C}, \mathcal{P}_0}$$

By Theorem 2.5.2,

$$\mathsf{NN}_{M_1'} \leq a(2\beta + 1)\mathsf{NN}_{M_1}$$

and Corollary 5.6.2 proves the result. □

5.7 Optimality of the coarsening sequence

The optimality of the coarsening sequence we produce is mostly due to the optimality property of the one-level coarsening algorithm, shown in the previous section. The one property that still has to be proven is that of local similarity. This section states the optimality theorem for the coarsening sequence, and shows the meshes are locally similar as well.

Theorem 5.7.1 (Good aspect ratio coarsening sequence) *Let M_0 be a mesh with smallest angle θ. The coarsening sequence $(M_1, ..., M_k)$ produced by Algorithm* MULTI_LEVEL_COARSEN_3 *has the following properties:*

1. **aspect ratio:** *There is a constant θ_1 depending on θ only such that for $1 \leq i \leq k$, the smallest angle of mesh M_i is bounded below by θ_1.*

2. **local similarity:** *There is a constant $\mathcal{I}$ depending on θ only such that for each $1 \leq i \leq k$, $\mathsf{el}_{M_{i+1}} \leq \mathcal{I}\mathsf{el}_{M_i}$.*

3. **size guarantee:** *Let $(M_1', ..., M_k')$ be any other good aspect ratio coarsening sequence of M_0 with smallest angle bound θ', then there exists a constant b depending on θ and θ' only such that*

$$|M_i| \leq b|M_i'| \quad 1 \leq i \leq k$$

Proof:

1. Direct corollary of Theorem 5.5.3.

2. Mesh M_i is generated using $f_{\mathsf{NN}_{\mathcal{P}_0},a,\mathcal{P}_0}$, and Mesh M_{i+1} is generated using spacing function $f_{\mathsf{NN}_{\mathcal{P}_0},2a,\mathcal{P}_0}$. By Lemma 5.4.2:
$$f_{\mathsf{NN}_{\mathcal{P}_0},2a,\mathcal{P}_0} \leq 2 f_{\mathsf{NN}_{\mathcal{P}_0},a,\mathcal{P}_0}$$
By Theorem 2.5.2:
$$\mathsf{NN}_{M_{i+1}} \leq c_1 f_{\mathsf{NN}_{\mathcal{P}_0},2a,\mathcal{P}_0} \leq 2 f_{\mathsf{NN}_{\mathcal{P}_0},a,\mathcal{P}_0} \leq c_2 \mathsf{NN}_{M_i}$$
By Theorem 2.2.9, for the appropriate setting of the constant $\mathcal{I}$:
$$\mathsf{el}_{M_{i+1}} \leq \mathcal{I} \mathsf{el}_{M_i}$$

3. By Theorem 5.6.3.

□

5.8 Efficient variations

So far, the basics of the function-based coarsening paradigm were presented and analyzed for correctness. This section addresses the issue of run-time complexity. The two main computational steps involved in creating a level in the coarsening sequence are:

1. computing the current level's coarsening function

2. computing a MIS of the intersection graph of the coarsening-function balls. Recall these are balls whose radii are a constant times the coarsening-function. The constant is β, the spacing constant.

As we discuss in this section, the coarsening spacing functions presented in previous sections can be computed at cost $O(n_0 \log n_0)$ per level, n_0 being the size of the mesh M_0. The second step, computing the MIS, can cost $O(n_0^2)$ work. This section presents a new set of coarsening functions together with a quadtree data structure which enable carrying out the computation of both steps at linear cost per level.

5.8.1 Computing coarsening functions

Recall that given an initial function f_0, a coarsening factor $\mathcal{C}$ and a set of points $\mathcal{P}$, the value of the $\mathcal{C}$-coarsening f_1 at point $p_i \in \mathcal{P}$ is defined as follows:
$$f_1(p_i) = \min_{p_j \in P} \mathcal{C} f_0(p_j) + ||p_i - p_j||$$

Thus the value of f_1 over the node set can be simply computed at cost $O(n^2)$. A better algorithm to compute f_1 is derived by noticing that the node p_j that gives $f_1(p_i)$ its value has the property that $\forall p_k \in P$:
$$\mathcal{C} f_0(p_j) + ||p_i - p_j|| \leq \mathcal{C} f_0(p_k) + ||p_i - p_k||$$
That is, p_i is contained in the additively weighted Voronoi cell of p_j. Fortune [30] shows how to apply the sweep-line technique to compute the additively weighted Voronoi diagram in $O(n_0 \log n_0)$, n_0 is the size of $\mathcal{P}$. The current level mesh size, n_i, can be much smaller. The techniques of this section show how to compute the coarsening function of level i at cost n_i.

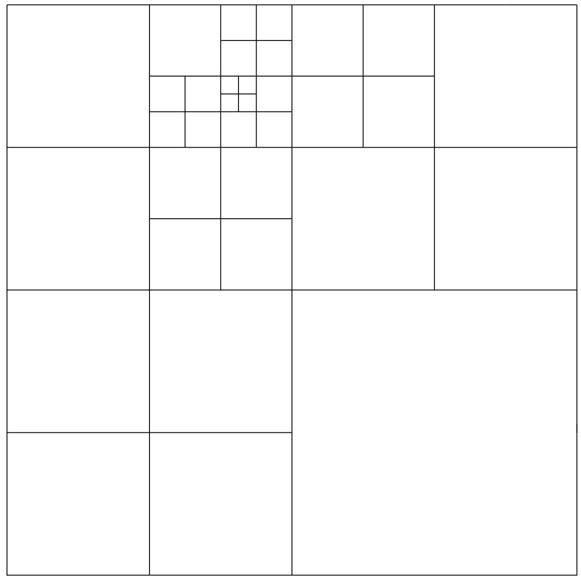


Figure 5.13: Leaf boxes of a balanced quadtree.

The coarsening functions $f_1, f_2, \ldots, f_k$ are optimal for coarsening, see Lemma 5.4.3. The optimality result carries over (up to a constant factor) to any sequence of coarsening functions $g_1, g_2, \ldots, g_k$ such that $a_1 g_i \leq f_i \leq a_2 g_i$. Rather than computing the original coarsening functions at a relatively high cost, an easier to compute but similar sequence can serve as well. The sequence of coarsening functions suggested here are called the quadtree coarsening functions. The rest of this section shows these functions approximate the spacing functions, yet are easy to compute.

Definition 5.8.1 (balanced quadtree [9]) *Two quadtree leaf boxes are neighboring if their intersection is a segment of non-zero length. A leaf box b is said to have a too-small neighbor if it neighbors a leaf box whose side length is smaller than half b's side length. A quadtree is called balanced if no leaf box b has a too-small neighbor. See Figure 5.13.*

Definition 5.8.2 (quadtree functions q_T and ℓ_T) *Let T be a balanced quadtree. For each box b of T, let $\ell(b)$ be the side length of b. Let $C = \{c_i\}$ be a set of points corresponding to the leaf boxes of T; each point is centered in its box.*

- *ℓ_T is a piece-wise constant function associated with T. Let b be a quadtree box of side length $\ell(b)$, x a point in b's interior, then assign $\ell(x) = \ell(b)/2$.*
- *q_T is a 1-Lipschitz function associated with T. It is defined as follows: For $c_i \in C$, let $q_T(c_i) = \ell(c_i)/2$, $\ell(c_i)$ being the side length of the leaf box c_i is in. For a point x in the domain, let*

$$q_T(x) = \min_{c_i \in C} q_T(c_i) + \|c_i - x\|$$

Lemma 5.8.3 *Let T be a balanced quadtree, then:*

1. *q_T is 1-Lipschitz. Furthermore, the value $q_T(x)$ of a point $x \in b$ is defined by a point $c \in C$ which is associated with an extended neighbor of b, i.e. a neighbor or a diagonal neighbor sharing a point with b.*

2. *$\ell_T(x) \leq q_T(x) \leq (1+\sqrt{2})\ell_T(x)$*

Proof:

1. q_T is 1-Lipschitz since it is a minimum over a set of 1-Lipschitz functions. Note that $q_T(c_i)$ is not affected by the minimum operation, since any other point c_j is at distance greater than $\ell(c_i)/2$ from c_i.

   The proof is by enumerating the small number of configuration of a balanced quadtree around the box b. However, to avoid the tedious proof this involves, assume momentarily that we defined $q_T(c_i) = \ell(c_i)/1000$ instead. If this definition is used, clearly when the min operation is performed to obtain the value of $q_T(x)$, the distance part of the terms dominates, and boxes with far centers can not effect the value of q_T. All the proofs below are true assuming this weaker definition of q_T, albeit with weaker constant factors.

2. Let b be a box of T, $x \in b$. Let c be b's center-point. By definition, $q_T(x) \leq \ell(b)/2 + \|b - x\| \leq (1+\sqrt{2})\ell(b)/2 = (1+\sqrt{2})\ell_T(x)$. The other direction is an implication of (1). The smallest the length of an extended neighbor can be is $\ell(b)/4$, hence $q_T(x) \geq \ell(b)/4 \geq \ell_T(x)/2$,

□

Theorem 5.8.4 *Given a general spacing function g on a square domain Ω, there exists a quadtree T such that its quadtree spacing function q_T approximates g, i.e. there exist two constants a_1 and a_2 such that:*

$$a_1 q_T \leq g \leq a_2 q_T$$

Proof: By Lemma 5.8.3, it suffices to show the result for ℓ_T rather than q_T. The proof is by construction:

1. Let Q be the start box Ω.

2. If $\ell(Q) > \min_Q g$, subdivide Q into 4 equal boxes of half the side length of Q.

   Repeat this step on all boxes, until none violate the condition.

3. Balance this partition by further subdividing any box that has a too-small neighbor.

By construction, $\ell_T \leq g$. The other inequality is now proven by induction. Let b be some box which is a leaf box by the end of step 2 of the algorithm, let p be its parent box. p was sub-divided so there is some point $x \in p$ such that $g(x) \leq \ell(p)$. Since b was not sub-divided, $\ell(b) = 0.5\ell(p) < g(y)$ for all points $y \in p$. By the distance between x and y, and the fact g is 1-Lipschitz, $g(y) \leq g(x) + \sqrt{3}\ell(p)$. In summary, by the end of step 2,

$$\ell(b) \leq g(y) \leq 2(\sqrt{3}+1)\ell(b) \qquad y \in b$$

In the balancing step, subdividing a box still maintains that for every point y in it $\ell(b) \leq g(y)$. For the other inequality, induct on the order in which the boxes were subdivided. The induction hypothesis is that for all boxes a subdivided before box b was split, $g(y) \leq C\ell(a)$ for all $y \in a$. Note

Algorithm: QUADTREE_COARSEN

1. Order the leaf boxes by their sizes, from smallest to largest.
2. For $i = 1$ to N:
   (a) If box b_i was coarsened already, do nothing.
   (b) If box b_i has a smaller neighbor, mark b_i coarsened.
   (c) Otherwise, replace box b_i and its other three siblings by their parent box. Mark all four siblings as coarsened.
3. Return the new balanced quadtree composed of all the coarsened boxes.

Figure 5.14: Coarsening the quad tree by a factor of two.

this assumption is true before splitting any box at step 2, for $C \geq 2(\sqrt{3}+1)$. Let b be a box about to be subdivided, c any of its children. Let a be the neighbor causing b to be split, i.e. $\ell(a) < 0.5\ell(b)$. Let x be any point of a. By the induction hypothesis, $g(x) \leq C\ell(a)$. The distance between x and some $y \in b$ is at most $\sqrt{3}(\ell(a) + \ell(b)) \leq (3/2)\sqrt{3}\ell(b)$. Therefore, $g(y) \leq g(x) + (3/2)\sqrt{3}\ell(b)) \leq C/2\ell(b) + 3/2\sqrt{3}\ell(b)$. Hence,

$$g(y) \leq (C/4 + 3/4\sqrt{3})\ell(c)$$

By using a C value such that $C/4 + \alpha 3/4\sqrt{3} \leq C$ the induction step is proven. □

coarsening quadtree function: To coarsen a quadtree function q_T, its associated balanced quadtree is coarsened. The coarsening method tries to replace four siblings that are all leaf boxes by their parent box. This is possible only if the four siblings do not have a small neighbor, hence coarsening proceeds from small leaf boxes to larger. See Figure 5.14

The following theorem shows that the quadtree functions q_{T_i} are similar to the optimal coarsening functions f_i:

Theorem 5.8.5 *Let T_i be the quadtree derived by applying Algorithm* QUADTREE_COARSEN *i times on T_0, then there exists a constant $a > 0$ such that $q_{T_0} > af_0$ implies $q_{T_i} > af_i$, where $f_{i+1} = f_{f_i,2}$.*

Proof:

It suffices to work with the set of functions ℓ_T since they are similar to q_T. Assume that $\ell_{T_i} > af_i$. It is now shown this implies $\ell_{T_{i+1}} > af_{i+1}$. To simplify the notation, let $s_1 = \ell_{T_{i+1}}$, $s_0 = \ell_{T_i}$, $g_0 = f_i$ and $g_1 = f_{i+1}$.

The claim is proven by induction on the size of the boxes in the tree T_i. The smallest box must have three siblings of equal size and hence was coarsened. Over the coarsened box, $s_1 \geq 2s_0 \geq 2ag_0 \geq ag_1$. This establishes the basis for induction. Let b_i be some box of the tree T_i. If b_i is coarsened, than by the same argument as for the smallest box, $s_1(\hat{b}) \geq ag_1$, where $\hat{b}$ is b's parent box. Otherwise, b_i is not coarsened. This implies that in the tree T_{i+1} b_i neighbors a smaller box b_s. By induction, $s_1 \geq ag_1$ for all $x \in b_s$. Let y be some point $y \in b_i$, and $x \in b_s$.

$$\begin{array}{llll} g_1(y) & \leq & g_1(x) + \|x-y\| & \text{by 1-Lipschitz of } g_1 \\ & \leq & g_1(x) + 10s_1(b_s) & \text{since } \|x-y\| \leq 10s_1(b_s) \\ & \leq & \frac{1}{a}s_1(b_s) + 10s_1(b_s) & \text{by induction assumption} \end{array}$$

$$
\begin{array}{lll}
\leq & s_1(b_i)(\frac{1}{2a}+5) & \text{since } s_1(b_i)=2s_1(b_s) \\
\leq & \leq \frac{1}{a}s_1(b_i) & \text{this is true whenever } a \leq 0.1.
\end{array}
$$

Theorem 5.8.6 *Given a mesh M, the cost of computing the quadtree coarsening functions sequence $q_1,\ldots,q_k$ is $O\left(\Sigma_{i=1}^{k}(P_i+Q_i)\right)$, where P_i is the size of the optimal mesh at level i, and Q_i is the number of function value queries at level i.*

Proof: Operations such as converting NN_M to a quadtree T_0 using the algorithm of Theorem 5.8.4 or ordering the quadtree boxes can be done in $O(P_0 \log P_0)$. The sum of the coarsening sequence mesh sizes, even in the case of quasi-uniform coarsening, is greater than $P_0 \log P_0$ so this one-time cost amortize to less than or equal to linear work per level. The work of coarsening each quadtree T_i in its turn is no more than P_i: coarsening the tree is linear in the number of its leaf boxes, and since the function q_i (and hence the tree boxes) is similar to an optimal coarsening function the number of tree boxes is $O(P_i)$. Finally, by Lemma 5.8.3, each query involves a minimum operation over the centers of neighboring boxes: a constant number of boxes per query. □

5.8.2 Generating the coarsening meshes

The cost of computing the MIS of the intersection graph can be $\Omega(n^2)$, especially when using large coarsening factors. The sequence of coarsening quadtrees generated in the previous subsection can be used as a data-structure to speed up these computations. It is intuitively clear that if a coarsened quadtree box contains many nodes of the original mesh $\mathcal{M}_0$, there is no need to consider them all in the mesh coarsening at this level. Rather, the following subset of the nodes can be considered:

Definition 5.8.7 ($P(T_i,k)$) *Let $\mathcal{M}_0=(P,E)$ be a good aspect ratio mesh, T_0 its associated quadtree. Let the quadtree T_i be the i-th level coarsening of T.*

Partition each leaf box b of T_i into 4^k smaller sub-boxes b_k of equal size. The point set $P(T_i,k)$ is constructed by picking an arbitrary point from $b_k \cap P$ when possible (this intersection can be empty).

Theorem 5.8.8 *Let $\mathcal{M}_0=(P,E)$ be a good aspect ratio mesh, with the bounded gaps property parameter $\mathcal{L}_0$. The point-sets $P(T_i,k)$ have the following properties:*

1. *$P(T_i,k)$ can be constructed at cost $|P(T_i,k)|$.*
2. *They can be constructed such that $P(T_{i+1},k) \subseteq P(T_i,k)$.*
3. *The point set $P(T_i,k)$ has the bounded gaps property with constant parameter $\mathcal{L}_1=1+2\mathcal{L}_0$, whenever $k \leq 0.5\log(6a\mathcal{L})$.*

Proof:

1. The cost of constructing $P(T_i,k)$ can be amortized in the cost of constructing the coarsening quadtree sequence, by marking or storing in the box one of the points whenever a new box is created.
2. Clearly, since the sub-boxes nest as well.
3. Let x be a point of Ω, consider a ball B of radius $(2\mathcal{L}_0+1)f(x)$ through x. Let y be a point on the radial from x, at distance $f(x)$ from x. Then, $f(y) \leq 2f(x)$, and a ball B_1 concentric with B, and of radius $2\mathcal{L}_0 f(x)$ must contain a point $p \in P$. Now, the rest of the proof proceeds

by setting k such that the box b_k containing p is wholly contained in B. Since $b_k \cap P$ is not empty, clearly $B \cap P(T_i, k)$ is not empty as well. The function value at p is at most $6\mathcal{L}_0 f(x)$ since $p \in B_1$. By the construction of the quadtree T, for some constant a the size of the box b containing p is $\ell(b) \leq a6\mathcal{L}_0 f(x)$. The distance from p to the boundary of B is $f(x)$, so it suffices to set k such that $4^k \leq 6a\mathcal{L}_0$.

□

When generating the coarsening mesh at level i, we can take the point set $P(T_i, k)$ as the generating point set rather than $\mathcal{P}_0$. The above theorem guarantees that $P(T_i, k)$ suffices since it has the bounded gaps property with respect to q_{T_i}. Furthermore, the sequence can be made node-nested using the same technique as before, that of proceeding from the coarse mesh to finer meshes, and giving priority to the coarser meshes nodes. Constructing the MIS of the conflict graph now becomes a linear operation. In particular, notice that β can be set so that the number of edges in the conflict graph is linear, and these edges can be found using a local search from each node.

5.9 Summary

The problem of mesh coarsening is that of creating a sequence of mesh coarsenings given an initial bounded aspect ratio mesh. All the meshes in the coarsening sequence should be of bounded aspect ratio. A mesh in the coarsening sequence is a smaller approximation of its preceding mesh.

This chapter addressed the problem of mesh coarsening theoreticaly, by first introducing the function-based coarsening approach, and then showing an algorithm based on this approach can be designed to be optimal in the following sense: (1) the sequence it constructs is a bounded aspect ratio sequence answering to the restriction of mesh coarsening and (2) is the smallest such sequence up to a constant factor in terms of both the sequence length and the size of the meshes in the sequence. Furthermore, the sequence can be constructed in time complexity linear in the total size of the sequence meshes.

A more practical version of the algorithm is developed and experimented with in following chapters. This chapter presents the theoretical algorithm which is the basis for later experimentation.

Chapter 6

Repeated Degradation of Iterative Coarsening

The problem of **repeated degradation** occurs when iteratively coarsening a mesh. The initial mesh is typically of good quality. A subset of the nodes is selected to create a coarser mesh. Coarsening algorithms usually degrade the quality of the coarser mesh somewhat; for example the aspect ratio of the elements becomes a little worse. As the process is repeated, the quality of the coarsening meshes eventually plunges below acceptable levels.

The previous chapter presented an approach to mesh coarsening that was not subject to repeated degradation. Note, however, that this approach was not iterative. The mesh of level i was generated using the point set of level M_0. This chapter discusses *iterative coarsening*, which means that the mesh at level $i+1$ is generated using only the information embedded in the mesh M_i of the previous finer level.

This chapter discusses the problem of repeated degradation, and the extent to which the function-based coarsening approach and the MIS-based coarsening approach can be adapted to resist degradation.

In this chapter meshes are classified into two categories: quasi-uniform meshes and graded meshes. This classification is necessary since the problem of repeated degradation is prevalent to different extents on these two mesh categories.

Section 6.1 discusses the problem of repeated degradation in the MIS-based approach. It shows that the problem can appear even in quasi-uniform meshes, and that the probability of repeated degradation occurring approaches one as the mesh size increases. However, it also shows that this effect is not prominent even on meshes with a few million nodes. In contrast, on general graded meshes repeated degradation of the MIS-method can be observed in practice even when the mesh has only a few thousand nodes, as the experiments of Chapter 8 show.

The function-based approach is tackled next. First, Section 6.2 shows that function-based coarsening can be adapted to coarsen quasi-uniform meshes in a manner theoretically guaranteed not to degrade the mesh quality. Section 6.3 generalizes this result to threshold function coarsening over general graded meshes. It is still an open question to try and further generalize the result to general coarsening functions over general graded meshes. The experiments of Chapter 8 show that in practice function-based coarsening is not subject to repeated degradation on all mesh types, quasi-uniform as well as graded.

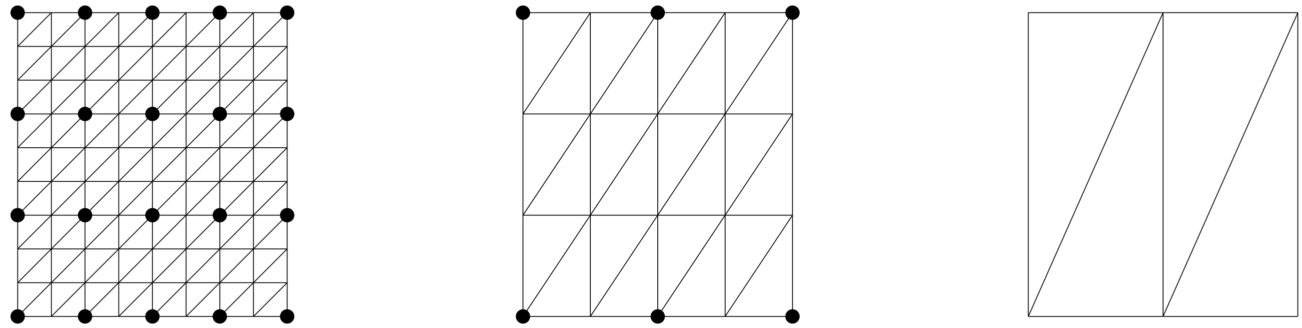


Figure 6.1: Repeated applications of MIS can degrade the aspect ratio.

6.1 Repeated degradation and MIS-based coarsening

Algorithm MIS_COARSEN of Section 5.2.1 is both simple and efficient. The mesh is viewed as a graph, and coarsened by picking a maximal independent set of the graph nodes to serve as the nodes of the coarser mesh. This coarse node set is then triangulated to form the coarse mesh. A popular triangulation method is the Delaunay triangulation. Given an initial mesh of size n, the MIS coarsens the mesh by a factor of 4 to 5 using $O(n)$ time complexity. Retriangulating the mesh using an $O(n \log n)$ Delaunay algorithm might be the costliest part of the algorithm. Randomized Delaunay triangulation methods, and bucketing techniques for uniform density point sets, reduce the complexity of the triangulation step. On the whole then, MIS-based coarsening can sometimes produce a coarsening sequence for an initial mesh sized n at a total run-time of $\sum_{i=1}^{\log_4^n} \frac{n}{4^i} = O(n)$.

The drawback of MIS_COARSEN is that it is subject to repeated degradation. Figure 6.1 gives an example where repeated degradation occurs using the MIS in conjunction with Delaunay triangulation over a quasi-uniform mesh. The initial mesh of Figure 6.1 is a grid-based triangular mesh. Subsequent coarsening meshes are constructed by selecting the MIS of previous meshes in the following way: every second node in selected from the x-dimension, and every third node from the y-dimension. This results in a coarsening sequence whose aspect ratio degrades as a function of $(2/3)^k$.

Let α_0 be the smallest angle of the level k coarsening mesh of our example. α_0 is:

$$\tan \alpha_0 = \left(\frac{2}{3}\right)^k$$

To understand what this formula implies in practice, consider the following table: the last row, S_k, is the smallest number of nodes necessary in order to arrive at a single angle of size α_0 at level k using that particular selection pattern.

level	1	2	3	4	5	6	7	8	9	10
α_0	33.6°	23.9°	16.5°	11.17°	7.5°	5.01°	3.34°	2.23°	1.49°	0.99°
S_k	6	36	216	1296	7778	46656	279936	1679616	10077696	60466176

Clearly, for repeated degradation to cause the appearance of one angle of bad quality the initial mesh must contain over a million nodes, and the choice of the maximal independent set is a deterministic, deliberately bad, choice. When constructing a MIS in a sequential fashion, the order in which the nodes are considered decides the MIS. A node is added to the MIS if it is considered before all of its graph neighbors. This order can be either fixed deterministically or randomized. The example above shows that deterministic MIS is subject to repeated degradation. What happens if we allow randomized orders?

The pattern of node selection of the example above of constant size, that is, to obtain an angle smaller than 1.5° a constant number of nodes of the initial mesh are used. If we have a large enough mesh, the pattern can be embedded at many areas of the mesh. The pattern is of constant size, so as the mesh size increases, the probability the pattern will arise approaches one. Note however that for reasonable mesh sizes this probability is too small to matter in practice.

To summarize, it can be theoretically proven that MIS is subject to repeated degradation even for uniform meshes, but for mesh sizes encountered in practice the MIS can produce high quality coarsening sequence. For graded unstructured meshes this is not the case, and Chapter 8 shows the MIS fails in practice as well as in theory.

6.2 Function-based coarsening of quasi-uniform meshes

This section shows that the parameters of function-based coarsening of quasi-uniform meshes can be set so that it is not subject to repeated degradation. We show this in a simple setting, and neglect the boundaries though a similar argument can be adapted for the lower dimensional case of the boundaries and their interaction with the interior. Consider the following simple coarsening scheme, using a coarsening factor $\mathcal{C} = 3$ and spacing constant $\beta = 2$:

Let $M_0 = (X_0, E_0)$ be the quasi-uniform input mesh.

Let $\mathcal{R}_0 = 0.5 \min_{(i,j)\in E} ||x_i - x_j||$. The set $\{\mathsf{B}(x, \mathcal{R}_0) | x \in X_0\}$ is a set of disjoint balls.

for $i = 1 \ldots \log |X|$

1. $\mathcal{R}_{i+1} = 3\mathcal{R}_i$.
2. Let $X_{i+1} \subseteq X_i$ be a maximal subset such that $\{\mathsf{B}(x, \mathcal{R}_{i+1}) | x \in X_{i+1}\}$ is a set of disjoint balls.
3. Let $M_{i+1} = (X_{i+1}, E_{i+1})$ be a Delaunay triangulation of X_{i+1}.

end for

Lemma 6.2.1 *Let $\mathcal{S}$ be a square of side length $L \geq 2\mathcal{R}_i$. Let $\mathcal{Q} = \{Q_1, \ldots, Q_8\}$ be the set of eight squares surrounding $\mathcal{S}$ as in Figure 6.2. A ball $\mathsf{B}(x, \mathcal{R}_i)$ centered in $\mathcal{S}$ can not intersect a ball $\mathsf{B}(y, \mathcal{R}_i)$ if y is located outside both $\mathcal{S}$ and $\mathcal{Q}$.*

Proof: Clear by the relative size of $\mathcal{R}_i$ to the size of the square, see Figure 6.2.

Lemma 6.2.1 can be used to show that it can not be the case that all nine squares are empty after coarsening: if the eight squares in $\mathcal{Q}$ are empty then a point in $\mathcal{S}$ must have survived the coarsening since it does not intersect any ball outside $\mathcal{Q} \cup \mathcal{S}$. We now use this lemma to show by induction that at level $i + 1$, a square of side length $6\mathcal{L}\mathcal{R}_{i+1}$ must contain a point of X_{i+1}. $\mathcal{L}$ is a constant greater than one, tuned to guarantee that every square of side length $6\mathcal{L}\mathcal{R}_0$ contains a point of X_0.

Theorem 6.2.2 *There exist a constant $\mathcal{L} \geq 1$ such that for each i, the mesh M_i computed at level i of the simple scheme contains no empty square of side length $L \geq 6\mathcal{L}\mathcal{R}_i$.*

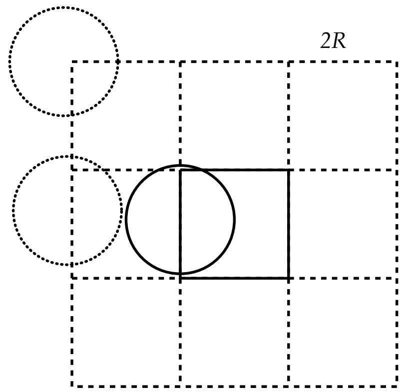


Figure 6.2: Lemma 6.2.1: a ball in the inner square is protected by the eight surrounding squares.

Proof: By induction:
Induction basis: there exists a constant $\mathcal{L} \geq 1$ such that every square of side length $6\mathcal{L}\mathcal{R}_0$ contains a point of X_0. This is because M_0 is a quasi-uniform good aspect ratio mesh.
Induction hypothesis: For all levels $l \leq i$, a square of side length $6\mathcal{L}\mathcal{R}_l$ contains at least one point of X_l.
Induction: *A square $\mathcal{S}_{i+1}$ of side length $6\mathcal{L}\mathcal{R}_{i+1}$ contains a point of X_{i+1}*
Proof: Recall that $\mathcal{R}_{i+1} = 3\mathcal{R}_i$. Divide the square $\mathcal{S}_{i+1}$ into nine equal squares, each of side length $2\mathcal{L}\mathcal{R}_{i+1} = 6\mathcal{L}\mathcal{R}_i$. By the induction assumption, each of the smaller squares contain a point of X_i. Assume (by way of contradiction) that none of the eight outer squares contains a point of the coarsened point set X_{i+1}. Using the induction hypothesis, the middle square contains at least one point $x \in X_i$. By Lemma 6.2.1, no ball centered outside the nine squares can intersect the ball $\mathsf{B}(x, \mathcal{R}_{i+1})$ centered at the middle square. Hence, the maximal set of disjoint balls will contain at least one point from the middle square, or a point in one of the eight surrounding squares. □

Corollary 6.2.3 *There exist a constant θ such that all the meshes M_i have smallest angle bounded below by θ.*

Proof: A circle of radius $3\sqrt{2}\mathcal{L}\mathcal{R}_{i+1}$ must contain a square of side length $6\mathcal{L}\mathcal{R}_{i+1}$. By Theorem 6.2.2, such a circle can not be empty. Therefore, using the empty circumcircle criterion, the radius of a circumcircle at level $i+1$ is at most $3\sqrt{2}\mathcal{L}\mathcal{R}_{i+1}$. Since edge length at that level is at least $2\mathcal{R}_{i+1}$ this gives a constant bounded radius-edge ratio:

$$\frac{R}{e} \leq \frac{3\sqrt{2}\mathcal{L}\mathcal{R}_{i+1}}{2\mathcal{R}_{i+1}} = \frac{3\mathcal{L}}{\sqrt{2}}.$$

Therefore, the smallest angle is also bounded by a constant. □

6.3 Threshold function coarsening

This section shows that coarsening iteratively, using a threshold coarsening functions sequence $\{t_i\}$ such that

$$t_{i+1}(x) = \max(t_i(x), \mathcal{C} \min_{\mathcal{P}_i} t_i)$$

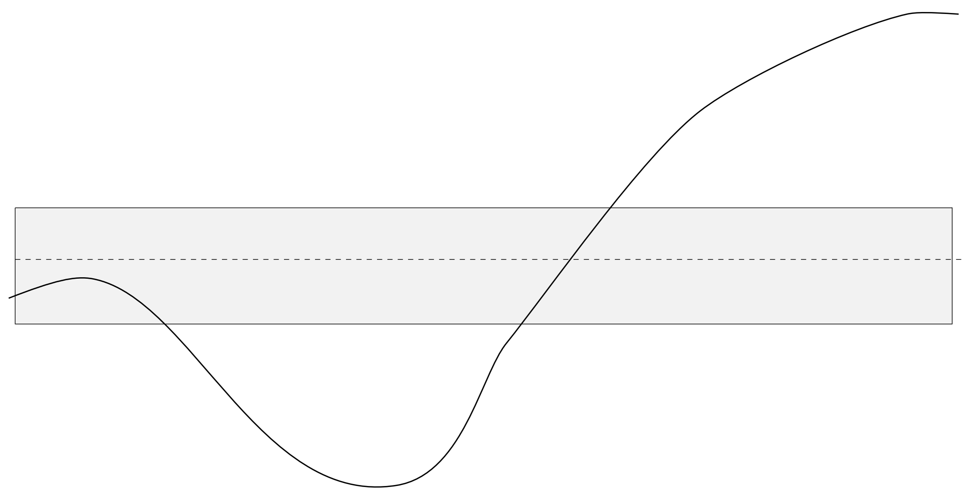


Figure 6.3: Threshold coarsening: the threshold value is marked by a dashed line, the initial function in a bold line. Only points whose function values lie in a region near the threshold value are affected by degradation. Theorem 6.3.3 shows this degradation is temporary.

is not subject to repeated degradation under a certain setting of $\mathcal{C}$, the coarsening factor, and of β, the spacing constant.

Note that coarsening of quasi-uniform meshes is in essence threshold coarsening. However, this section generalizes the previous section since the mesh itself is not assumed to be quasi-uniform. New techniques are necessary for this case. While performing threshold coarsening, the domain points can be classified into three categories: (1) points with large function values such that they, and a whole area around them, are not affected at all by the threshold coarsening (2) points with very small values such that the threshold coarsening bypassed them a few iteration ago and from their standpoint the coarsening seems like quasi-uniform coarsening, and (3) points whose value is close to the threshold value. Figure 6.3 highlights this category. The intuition behind the proofs below is that categories (1) and (2) are not affected much by the coarsening, and that their quality does not degrade, whereas points of the third category are affected and their quality does degrade, however they quickly pass into the second category and hence the amount of degradation is bounded and temporary. The details of the proof are quite involved, and presented for the simple non-boundary case.

The following auxiliary lemma shows that if the f-balls of two points intersect, then the function values at the points are close:

Lemma 6.3.1 *Let q and p be two points in a domain Ω, such that their f-balls intersect, that is,* $\mathsf{B}(p, \frac{f(p)}{\beta}) \cap \mathsf{B}(q, \frac{f(q)}{\beta}) \neq \emptyset$. *Then:*

1. $f(q) \geq \frac{\beta-1}{\beta+1} f(p)$

2. $\|p-q\| \leq \frac{2f(p)}{\beta-1}$

Proof:

1. By the Lipschitz condition, $f(q) \geq f(p) - d$, d being the distance between p and q. Since the balls intersect, $f(p) + f(q) > \beta d$, or $d \leq (f(q) + f(p))/\beta$. Combining the two inequalities, $f(q) > f(p) - (f(q) + f(p))/\beta$. The result is a rearrangement of the last inequality.

2. If two f-ball intersect, the distance between the corresponding points must be smaller than:

$$\frac{f(p)}{\beta} + \frac{f(q)}{\beta} \leq \frac{f(p)}{\beta}(1 + \frac{\beta+1}{\beta-1}) = \frac{2f(p)}{\beta-1}$$

□

The following lemma shows that under certain conditions the coarsening process actually decreases the size of the gap at a point.

Lemma 6.3.2 *Let $\mathcal{P}_0$ be a point set β-spaced by a spacing function t_0, with gaps bounded by $2\mathcal{L}$. Let t_1 be the threshold coarsening function:*

$$t_1(x) = \max(t_0(x), \mathcal{C} \min_{\mathcal{P}_0} t_0)$$

Let $\mathcal{P}_1$ be the result of function coarsening $\mathcal{P}_0$ using t_1. If x is a point such that $t_1(x) \geq 3t_0(x)$, and $\beta \geq 8\mathcal{L}$ then the gaps at x are bounded by $\mathcal{L}$.

Proof: Let B be some $\mathcal{L}$ gap at x, i.e. $B = \mathsf{B}(c, R)$ is a ball with center c passing through x of radius $R = \mathcal{L}t_1(x)$. The goal is to show that B must contain a point $p \in \mathcal{P}_1$.

Let $B_1 = \mathsf{B}(c, R - \ell)$ be another ball centered in B, for $\ell = \frac{2t_1(x)}{\beta-1}$. Let y be the point of B_1 intersecting the radial from x to c.

We now show that the $2\mathcal{L}$-gap at y with respect to t_0 is contained in B_1. By the Lipschitz condition, $t_0(y) \leq t_0(x) + \ell$. The radius of the $2\mathcal{L}$ gap at y is therefore at most $2\mathcal{L}t_0(x) + 2\mathcal{L}\ell$. This is contained in B_1 if $2\mathcal{L}t_0(x) + 2\mathcal{L}\ell \leq R - \ell = \mathcal{L}t_1(x) - \ell$. Reorganizing the terms, the gap is contained in B_1 if:

$$2\mathcal{L}t_0(x) \leq t_1(x)\left(\mathcal{L} - \frac{2(1+2\mathcal{L})}{\beta-1}\right)$$

which is clearly true when $\beta \geq 8\mathcal{L}$ (assuming also $\mathcal{L} \geq 1$) since $t_1(x) \geq 3t_0(x)$.

The fact that y's level zero gap is contained in B_1 means there is a point $p \in B_1 \cap \mathcal{P}_0$. This point is at a distance of at least $\frac{2t_1(x)}{\beta}$ from the exterior of B. If $p \in \mathcal{P}_1$, then we are done since B is not empty. Otherwise, there must exist some point $q \in \mathcal{P}_1$ sharing a conflict graph edge with p that is responsible for eliminating p from $\mathcal{P}_1$. If $q \in B$ we are done, so assume q is exterior to B.

Note that either $t_1(p) > t_0(p)$ or $t_1(q) > t_0(q)$, otherwise they can not share a conflict graph edge with respect to t_1 since they do not share one with respect to t_0. Without loss of generality, assume it is q. Because of the threshold function coarsening this implies $t_1(q) = \min t_1$. But it is also true that $t_1(x) = \min t_1$ so $t_1(q) = t_1(x)$. By Lemma 6.3.1 the t_1-balls of p and q can intersect only if the distance between them is smaller than $\frac{2t_1(q)}{\beta-1} = \frac{2t_1(x)}{\beta-1} = \ell$, a contradiction. □

The following theorem shows the parameters of a threshold function-based coarsening algorithm can be set to guarantee no repeated degradation occurs. Recall that function-based coarsening proceeds by computing the function values $f(p)$ at a set of nodes $\mathcal{P}$, drawing f-balls around nodes with radii of size $f(p)/\beta$, forming the conflict graph by drawing an edge between any two nodes whose balls intersect, and finally, selecting a maximal independent set of the conflict graph to be the coarsened set of nodes.

Theorem 6.3.3 *Let $\mathcal{P}_0$ be a point set β-spaced by a spacing function t_0, with gaps bounded by $\mathcal{L}$. Define a sequence of threshold coarsening functions in the following way:*

$$t_{i+1}(x) = \max(t_i(x), \mathcal{C} \min_{\mathcal{P}_i} t_i)$$

Let $\mathcal{P}_{i+1}$ be the result of function-based coarsening of $\mathcal{P}_i$ using t_{i+1}.

For each level i, define two sets $\mathcal{B}_i$ and NB_i:

- *$\mathcal{B}_i$: points $x \in \Omega$ whose gaps are bounded by $\mathcal{L}$ with respect to $\mathcal{P}_i$ and t_i.*
- *NB_i: points $x \in \Omega$ with the following properties:*
  - *(a) their gaps are bounded by $2\mathcal{L}$ with respect to $\mathcal{P}_i$ and t_i but have at least one empty $\mathcal{L}$-gap, that is, they have an unbounded $\mathcal{L}$-gap with respect to t_i.*
  - *(b) $t_i(x) < 8 \min t_i$.*

If $\mathcal{C} \geq 24$ and $\beta \geq 8\mathcal{L}$ then:

1. $\Omega = \mathcal{B}_i \cup NB_i$
2. *if* $x \in NB_{i-1} \Rightarrow x \in \mathcal{B}_i$
3. *if* $x \in NB_i \Rightarrow x \in \mathcal{B}_{i-1}$

Proof: Note, that property (1) implies the gaps are always bounded by $2\mathcal{L}$, and therefore this threshold coarsening scheme is not subject to repeated degradation.

The proof proceeds by induction. At step zero $\Omega = \mathcal{B}_0$ since the input point set $\mathcal{P}_0$ is of bounded $\mathcal{L}$-gaps with respect to t_0. Assume that the induction hypotheses (1)-(3) are true as stated for level i. We now prove them for level $i+1$.

1. First, we show that if $x \in \mathrm{NB}_i$ then $x \in \mathcal{B}_{i+1}$. By definition, points of NB_i have $t_i(x) < 8 \min t_i$. If $\mathcal{C} > 24$, then in the next iteration $t_{i+1}(x) > 3t_i(x)$, and hence by Lemma 6.3.2 and the induction hypothesis, $x \in \mathcal{B}_{i+1}$.
2. Now, let x be some point in $\mathcal{B}_i$. If all the $\mathcal{L}$ gaps through x with respect to t_{i+1} are non-empty then $x \in \mathcal{B}_{i+1}$. Otherwise, there exists a gap $B = \mathsf{B}(c, \mathcal{L}t_{i+1}(x))$ passing through x that contains no point of $\mathcal{P}_{i+1}$. In that case, we show that $x \in \mathrm{NB}_{i+1}$ by showing that the following two facts hold:
   (a) **Gaps bounded by $2\mathcal{L}$:** to show this property, it suffices to show that the ball B_1 whose center is on the line extending the radial from x to c, and whose radius is $2\mathcal{L}t_{i+1}(x)$, contains some point of $\mathcal{P}_{i+1}$.
   Consider Figure 6.4. It shows the $\mathcal{L}$-gap B through x, centered at c and drawn in bold, and the $2\mathcal{L}$-gap B_1. Let y be a point at distance $\ell = \frac{\mathcal{L}}{\mathcal{L}+1} t_{i+1}(x)$ from x. Let B_2 be an $\mathcal{L}$-gap through y. $t_{i+1}(y) \leq t_{i+1}(x) + \ell$, therefore the radius of the gap at y is at most $\mathcal{L}t_{i+1}(x)(1 + \mathcal{L}/(\mathcal{L}+1))$. The important thing to notice about the setting of ℓ is that it enables us to nestle the gap B_2 within the gap B_1 such that the distance between the outside of B_1 and the interior of B_2 is at least ℓ.

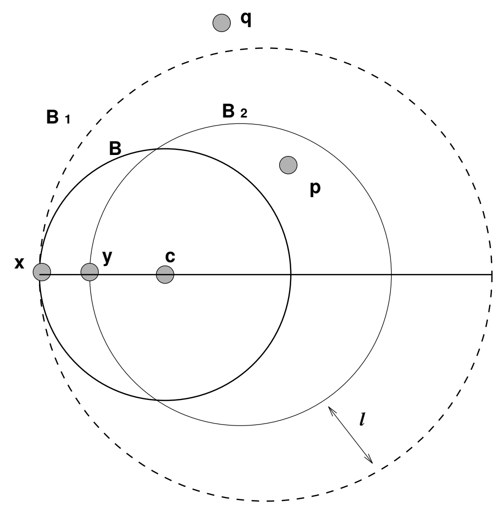


Figure 6.4: Illustration for Theorem 6.3.3

If $y \in \mathcal{B}_{i+1}$ then B_2 contains a point $p \in \mathcal{P}_{i+1}$. As a consequence, so does B_1 and we are done. Otherwise, y is either in NB_i or in $\mathcal{B}_i$. If $y \in \mathrm{NB}_i$ then by the first part of this induction $y \in \mathcal{B}_{i+1}$, a contradiction. Therefore, $y \in \mathcal{B}_i$. B_2 then contains a point $p \in \mathcal{P}_i$ such that $p \notin \mathcal{P}_{i+1}$. The fact p is not in $\mathcal{P}_{i+1}$ implies there exists a point $q \in \mathcal{P}_{i+1}$ such that p and q share a conflict graph edge. We show that B_1 is not empty by showing $q \in B_1$. Assume, by way of contradiction, that q is outside B_1. The distance between p and q is then at least ℓ. Since both $p, q \in \mathcal{P}_i$, their t_i-balls do not intersect. Their t_{i+1}-balls can intersect only if either p or q increased their coarsening function value. Without loss of generality, assume $t_{i+1}(q) \neq t_i(q)$. Because of the nature of threshold coarsening this means

$$t_{i+1}(q) = \min t_{i+1} \leq t_{i+1}(x)$$

Recall that the distance between p and q is at least $\ell = \frac{\mathcal{L}}{\mathcal{L}+1} t_{i+1}(x) \geq \frac{\mathcal{L}}{\mathcal{L}+1} t_{i+1}(q)$. By Lemma 6.3.1, the distance between them must be smaller then $\frac{2}{\beta-1} t_{i+1}(q)$. If $\beta \geq 2$ this is a contradiction and hence $q \in B_1$ and we are done.

(b) $\mathbf{t_{i+1}(x) < 8 \min t_{i+1}}$: Again, we consider the gap $B = \mathsf{B}(c, \mathcal{L} t_{i+1}(x))$ through x that became empty in this coarsening round. Let $p \in \mathcal{P}_i \cap B$ be the point of $\mathcal{P}_i$ closest to the center c. Let r_p be the distance between p and c. Let $R = \mathcal{L} t_{i+1}(x)$ stand for the radius of B. We argue by the following cases:

First, assume $r_p \leq R/2$. Since p is not in $\mathcal{P}_{i+1}$ it shares a conflict graph edge with a point $q \in \mathcal{P}_{i+1}$. Letting ℓ stand for the distance between p and q this means that $t_{i+1}(p) + t_{i+1}(q) > \beta\ell$. Note that $\ell > R/2$. By the Lipschitz condition,

$$2t_{i+1}(p) + \ell \geq t_{i+1}(p) + t_{i+1}(q) > \beta\ell$$

so that

$$t_{i+1}(p) > \frac{\ell(\beta-1)}{2} > \frac{R(\beta-1)}{4} = \frac{\mathcal{L} t_{i+1}(x)(\beta-1)}{4}$$

Hence:

$$t_{i+1}(x) < \frac{4t_{i+1}(p)}{\mathcal{L}(\beta - 1)}$$

Now assume $r_p > R/2$. Recall that p is picked as the point of $\mathcal{P}_i$ closest to c. By the induction assumption, the level i gap at p is bounded by $2\mathcal{L}$. This gap radius must be larger than the distance from p to c otherwise a point in the gap of p is closer to c than p is, leading to a contradiction. Therefore $2\mathcal{L}t_i(p) \geq R/2 = \mathcal{L}t_{i+1}(x)/2$, or using the fact that $t_{i+1}(p) \geq t_i(p)$,

$$t_{i+1}(x) \leq 4t_{i+1}(p)$$

Therefore, in either case we have that $t_{i+1}(x) \leq 4t_{i+1}(p)$. Since p and q are both in $\mathcal{P}_i$ but $p \notin \mathcal{P}_{i+1}$, the coarsening function values of either p or q increased. We assume q is the one whose value increased since this gives worse bounds. Therefore, $t_{i+1}(q) = \min t_{i+1}$. Since p's ball intersects q's (they share a conflict graph edge) Lemma 6.3.1 implies $t_{i+1}(p) \leq t_{i+1}(q)(\beta+1)/(\beta-1)$. When $\beta \geq 3$ this means: $t_{i+1}(x) \leq 4t_{i+1}(p) \leq 8 \min t_{i+1}$.

□

6.4 Summary

This chapter discussed *iterative coarsening*, which means that the mesh at level M_{i+1} is generated using only the information embedded in the mesh M_i of the previous finer level. Iterative coarsening is particularly vulnerable to the problem of repeated degradation. Coarsening algorithms usually degrade the quality of the coarser mesh somewhat; for example, the aspect ratio of the elements becomes a little worse. As the process is repeated, the quality of the coarsening meshes eventually plunges below acceptable levels.

The results of this chapter addresses the iterative application of two coarsening methods: MIS-based coarsening and function-based coarsening. It should be noted that MIS-based coarsening must always be applied iteratively, whereas function-based coarsening can be applied non-iteratively as well and thus bypass the question of repeated degradation altogether.

For quasi-uniform meshes, the following results have been shown:

- MIS-based coarsening is subject to repeated degradation. An example has been provided to show that deterministic selection of the MIS can result in a coarsening sequence with geometrically increasing aspect ratios. Furthermore, it has been shown that for randomly selected MIS repeated degradation can occur with probability approaching one as the initial mesh size grows. However, it is also pointed out that the mesh sizes at which repeated degradation takes effect are too large to be of practical concern currently.

- Function-based coarsening has been theoretically shown to be resistant to repeated degradation.

For general unstructured meshes, the MIS-based coarsening fails both theoretically and in practice, as the experiments of chapter 8 to follow show. Function-based coarsening performs well in practice both over quasi-uniform meshes and over general unstructured meshes.

The question of theoretical guarantees for iterative function-based coarsening is still open. This chapter generalizes the guarantees against repeated degradation only to the case of threshold function coarsening of general unstructured meshes, a step away from general function coarsening of these meshes.

Chapter 7

Developing a Practical Coarsening Algorithm

The problem of mesh coarsening was introduced in Chapter 5: given a good aspect ratio mesh construct a sequence of respectively coarser meshes $\{M_0, M_1, \ldots, M_k\}$. Also, a new approach called function-based coarsening, was described and analyzed. The function-based approach generates mesh M_i, the i-th mesh of the coarsening sequence of the input mesh M_0, using the following two main computational steps:

- **coarsening function computation:** computing the value of the coarsening function f_i at the nodes of the generating node set (the generating node set is the set from which the nodes of M_i are selected).

- **nodes selection:** selecting a subset of the generating node set, such that the subset is well-spaced with respect to the coarsening function f_i. Mesh M_i is then a (constrained) Delaunay triangulation of the selected nodes.

Two types of coarsening functions were introduced in previous chapters: the standard coarsening functions, and quadtree coarsening functions. Using the quadtree coarsening functions, it is feasible to generate mesh M_i at cost $O(n_i)$, linear in n_i – the number of nodes in mesh M_i.

The quadtree based coarsening algorithm does not carry the same promise of simplicity that makes the MIS-based algorithm so appealing; quadtrees are often described in the mesh generation literature as resulting in unnecessarily large meshes (though size-optimal up to a constant factor [9]). When pushing technology to the limit and using meshes whose size totter close to the available memory, space is at a premium and large constant factors can mean "unusable".

An algorithm that combines the simplicity of the MIS approach with the aspect ratio quality of the function based approach (at least in practice) would have the best of both worlds; this chapter develops such a algorithm.

To derive a more practical algorithm, two ideas for modifying the theoretical coarsening algorithm of Chapter 5 are used:

1. **fine-to-coarse coarsening order:** in this approach, the generating node-set of mesh M_{i+1} is the nodes of mesh M_i, and the meshes of the sequence are generated in a finer mesh to coarser mesh order. This iterative method of coarsening is a very natural way to generate the coarsening sequence.

The problem is that this order is subject to repeated degradation. Mesh M_0 provides a good set of nodes to sample from which the nodes of M_1 can be sampled, mesh M_1 in turn provides a slightly worse node set to sample from; the problem compounds as the levels increase. In particular, see the discussion of Chapter 6.

This problem is circumvented if the nodes of M_0 serve as the generating set for every mesh in the coarsening sequence. For this case, Chapter 5 showed theoretical guarantees for the coarsening sequence quality. However, using M_0 as the generating set causes obvious efficiency problems (which were solved using quadtrees) since n_0 can be much greater than n_i. Also, in order for the sequence to be node-nested, a coarse-to-fine order must be employed, and the point set of the coarser mesh is then used as the kernel of the finer mesh below it in the coarsening sequence.

The first idea towards simplification is employing the more natural fine-to-coarse order, even at the cost of loosing the theoretical guaranty against repeated degradation. Chapter 6 makes some theoretical progress in showing that repeated degradation does not occur in function-based coarsening employed in this more natural fine-to-coarse iterative order. In particular, threshold-function coarsening is not subject to repeated degradation. It is currently unclear if the theory can be extended to cover general coarsening functions, but in practice the effects of repeated degradation can be neglected for function-based coarsening, even for very graded, unstructured meshes. The experiments of Chapter 8 substantiate this claim.

The fine-to-coarse order impacts the efficiency of both phases of the algorithm. Letting $n_i = |M_i|$:

- coarsening functions computations: laying quadtree technology aside, the coarsening functions can be computed using an additively weighted Delaunay diagram at cost $O(n \log n)$ (see Section 5.8.1), where n is the size of the generating node set. In the coarse-to-fine order the cost per level is $O(n_0 \log n_0)$, whereas in the fine-to-coarse order the cost for level $i+1$ is $O(n_i \log n_i)$. Note that the diagram can not be re-used among levels since the additive constant, i.e. the coarsening factor times the previous function value at the node, changes from level to level.
- nodes selection: nodes are selected as a maximal independent set of the intersection graph of the balls centered at the nodes of the generating set. The balls radii depend on the function value at the node and the coarsening constant. For coarse-to-fine construction, the intersection graph can be as large as $O(n_0^2)$. In a fine-to-coarse order, each mesh is a coarsening by a constant factor of the previous mesh. The intersection graph in that case can be shown to be linear (see section 7.3 below) and easily computable in $O(n_i)$ at level $i+1$. This is the most important computational benefit of the fine-to-coarse construction order.

2. **using the weighted graph metric:** Using the underlying *weighted graph metric* instead of the Euclidean metric enables the development of a simple and efficient algorithm for the first phase, the phase of computing the new coarsening function values. Sections 7.1 and 7.2 show that the theoretical results are still valid for the weighted graph metric, and that very simple and efficient shortest paths algorithm can be used. Furthermore, the weighted graph metric has the benefit that it very naturally enables computing *constrained* distances, i.e. geodesic distances in the presence of obstacles such as holes.

The benefits are quite substantial: the subroutine necessary for the Euclidean metric is additive weight constrained Voronoi diagram; the graph metric requires only Dijkstra's shortest paths. Whereas both subroutines are of equal asymptotic complexity, Dijkstra's is by far the simpler (and faster) routine.

The combination of these two simplification ideas results in a very simple $O(n_i \log n_i)$ algorithm to generate mesh M_{i+1}, whose main subroutine is Dijkstra's shortest paths algorithm. The next chapter, Chapter 8, shows that this algorithm is both efficient *and* produces high quality coarsening sequences in practice. No repeated degradation is observed during the experiments, and the resulting coarsening sequence is of good quality.

Section 7.1 discusses the weighted graph metric, and its appropriateness for the problem at hand. Section 7.2 shows how to compute the graph metric coarsening functions efficiently using Dijkstra's algorithm. Section 7.3 shows that selecting the points according to the coarsening function can be done efficiently when the coarsening factor is a constant, leading the discussion towards the iterative graph-based coarsening method. Section 7.4 discusses the handling of the boundaries in function-based coarsening.

7.1 Weighted graph metric

The metric used in this chapter is the *weighted graph metric* of the Delaunay triangulation. In this metric, the weight of the edge between two nodes is their Euclidean distance, whereas the distance between two nodes is the smallest sum of edge weights of a graph path connecting them.

Definition 7.1.1 (edge weight) *The weight $w_{(i,j)}$ of edge $e = (i,j)$ in the mesh M is $||x_i - x_j||$, the Euclidean distance between the corresponding mesh nodes.*

Definition 7.1.2 (weighted graph distance $d_G(.,.)$) *Let $G = (X, E)$ be an embedded graph. Let $P(i,j)$ be a graph path between nodes x_i and x_j, $w(P(i,j))$ the weight of the path, a sum of the path edge weights. The weighted graph distance between nodes x_i, x_j is:*

$$d_G(x_i, x_j) = \min_{P(i,j)} w(P(i,j))$$

Definition 7.1.3 (dilation) *The dilation of an embedded graph $G = (X, E)$ is*

$$\mu_G = \max_{x_i, x_j \in X} \frac{d_G(x_i, x_j)}{||x_i - x_j||}$$

Chew [20] showed that the *rectilinear* Delaunay triangulation is of dilation bounded by $\sqrt{10}$. He also gave a lower bound on the dilation of the standard Euclidean Delaunay triangulation: placing points along the circumference of a circle, the dilation can be shown to be at least $\Pi/2 \approx 1.570$. Dobkin, Friedman and Supowit [25] were the first to provide a bound on the dilation of the Euclidean Delaunay triangulation. They showed it is at most $(1+\sqrt{5})\Pi/2 \approx 5.083$. This bound has later been improved to $2\Pi/(3\cos(\Pi/6)) \approx 2.418$ by Keil and Gutwin [39]. Eppstein [28] in his survey "Spanning Trees and Spanners" describes this and other related work in the larger context of *spanners*, graphs of low dilation.

The dilation of the Euclidean Delaunay triangulation is therefore known to be $1.570 \leq \mu \leq 2.418$. These bounds are valid for general Delaunay triangulations; the meshes we are interested in are of

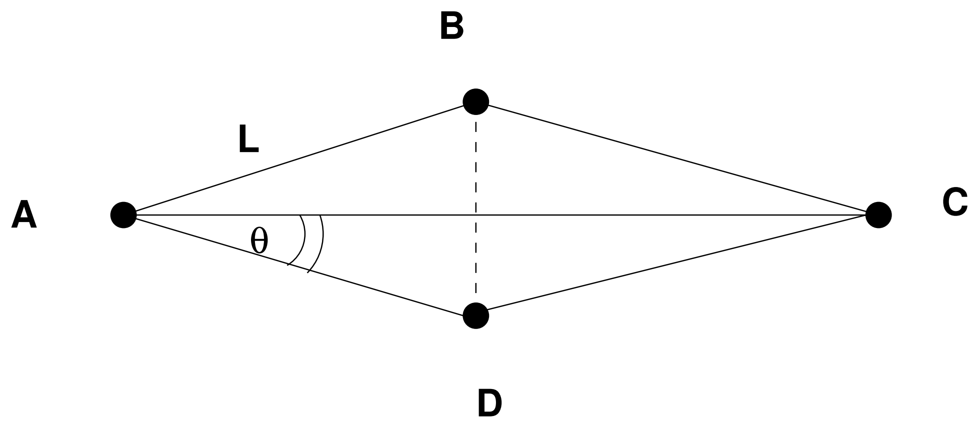


Figure 7.1: The graph distance between B and D is $2L$, and the relationship between the graph metric and Euclidean metric is $\|B - D\| = 2L \sin\theta$. The dilation is therefore at least $\mu(\theta) = \frac{1}{\sin\theta}$. This value is larger than the bounds for Delaunay triangulations. For example, $\mu(10°) = 5.75$, and $\mu(5°) = 11.47$.

good aspect ratio, and likely to be of smaller dilation. It is possible to bound the dilation by using only the aspect ratio of mesh, rather than its Delaunay graph properties. These bounds then depend on θ, and are weaker. See Figure 7.1.

The boundaries of the mesh necessitate redefining the dilation as the ratio between the graph metric and the *constrained* Euclidean metric. In the constrained Euclidean metric, the shortest path between two points is composed of boundary segments and segments that are interior to the domain, i.e. the geodesic distance. We conjecture Keil and Gutwin's proof can be extended to that case.

The extent to which graph metric based spacing functions approximate the Euclidean spacing functions also depends on the dilation μ:

Lemma 7.1.4 *Let $M = (X, E)$ be a mesh. Let f be set of values associated with the mesh nodes X. (f can be thought of as a spacing function, but we do not impose Lipschitz restrictions on the values). Let f_M be a coarsening of f by a coarsening parameter $\mathcal{C}$, using the underlying graph metric of M. i.e.*

$$f_M(x_i) = \min_{x_j \in X} \{\mathcal{C} f(x_j) + d_G(i, j)\}$$

Let $f_{\mathcal{E}}$ be coarsening by a factor of $\mathcal{C}$ using the Euclidean metric, i.e.

$$f_{\mathcal{E}}(x_i) = \min_{x_j \in X} \{\mathcal{C} f(x_j) + \|x_i - x_j\|\}$$

then:

$$f_{\mathcal{E}}(x_i) \le f_M(x_i) \le \mu f_{\mathcal{E}}(x_i) \quad \forall x_i \in X$$

Proof: f_M is a minimum over terms of the form $\mathcal{C} f(x_i) + d_M(i, j)$, each of which is larger than the corresponding term of $f_{\mathcal{E}}$, since $d_M(i, j) \ge ||x_i - x_j||$. Hence, $f_M \ge f_{\mathcal{E}}$.

Assume the value $f_M(x_i)$ depends on node x_j, where as the value of $f_{\mathcal{E}}(x_i)$ depends on node x_k. Then, $f_M(x_i) \le \mathcal{C} f(x_k) + d_M(i, k) \le \mathcal{C} f(x_k) + \mu ||x_i - x_k|| \le \mu f_{\mathcal{E}}(x_i)$. □

Corollary 7.1.5 *f_M is μ-Lipschitz ($\frac{1}{\mu} f_M$ is 1-Lipschitz).*

Theorem 7.1.6 (correctness of graph metric coarsening functions) *Let $f_{M,\mathcal{C}}$ be a graph metric coarsening function over mesh M, let β_M be its spacing constant. Let $f_{\mathcal{E},\mathcal{C}}$ be a Euclidean coarsening spacing function over the same mesh, and with the same coarsening factor. Let $\beta_{\mathcal{E}}$ be its spacing constant. If $\beta_M \geq \mu\beta_{\mathcal{E}}$ then:*

1. **(conflict-graph)** *Let $CG_M = (V, E)$ be the graph metric conflict graph. V is set to the nodes of M, and the edge set E contains all the edges of the form:*

$$\{(i,j)| \quad f_M(x_i) + f_M(x_j) \geq \beta_M d_M(i,j)\}$$

   then CG_M is a subgraph of $CG_{\mathcal{E}}$ (See Definition 5.3.4).

2. **(bounded gaps)** *Any MIS of CG_M has the bounded gaps property with respect to f_M with gap parameter $\mathcal{L}_1$.*

3. **(aspect ratio)** *A Delaunay triangulation of any MIS of CG_M is of bounded aspect ratio.*

Proof:

1. To show $\mathrm{CG}_M \subseteq \mathrm{CG}_{\mathcal{E}}$ it suffices to show that whenever:

$$f_M(x_i) + f_M(x_j) \geq \beta_M d_M(i,j)$$

   then it is also the case:

$$f_{\mathcal{E}}(x_i) + f_{\mathcal{E}}(x_j) \geq \beta_{\mathcal{E}}\|x_i - x_j\|$$

   But by Lemma 7.1.4:

$$\mu(f_{\mathcal{E}}(x_i) + f_{\mathcal{E}}(x_j)) \geq f_M(x_i) + f_M(x_j) \geq \beta_M d_M(i,j) \geq \beta_M\|x_i - x_j\|$$

   and the result follows since $\beta_M \geq \mu\beta_{\mathcal{E}}$.

2. Looking at the proof of Theorem 5.5.3, and using part (1), it is clear that the same bounded gaps result follows.

3. Let p be a Delaunay point. By part (3), the radius of an empty Delaunay ball through p is at most $\mathcal{L}_1 f_M(p)$. The shortest edge at p is however at least $f_M(p)/\beta_M$, and hence the radius-edge ratio is bounded. A bound on the aspect ratio then follows.

□

7.2 Computing the coarsening function

The problem of computing the graph metric coarsening functions can be stated as follows. Given

1. a graph $G = (P, E)$,

2. a set of old spacing function values at the nodes, $\{f_0(p_1)\ldots f_0(p_n)\}$, and

3. a coarsening factor $\mathcal{C}$,

compute the graph metric coarsening function at the nodes P:

$$f_1(p_i) = \min_{p_j \in P} \mathcal{C} f_0(p_j) + d_G(x_i, x_j)$$

These values can be computed simply and efficiently using an $O(n \log n)$ algorithm. The problem is transformed into a shortest paths problem, on the following graph:

Definition 7.2.1 ($\hat{\mathbf{G}}$) *Given a graph $G = (P, E)$ define the graph $\hat{G} = (\hat{P}, \hat{E})$ by:*

- **point-set:** $\hat{P} = P \cup \{p^*\}$.
- **edges:** $\hat{E} = E \cup \{(p^*, p_i) | \quad p_i \in P\}$.
- **edge-weights:**

$$w(e_{(i,j)}) = \begin{cases} ||x_i - x_j|| & e_{(i,j)} \in E \\ \mathcal{C} f_0(p_i) & e = (p^*, p_i) \end{cases}$$

Theorem 7.2.2 *$f_1(p_i)$ is equal to the shortest graph distance between p^* and p_i in the graph $\hat{G}$.*

Proof: Any path from node p^* to node p_i contains at most one edge of the form (p^*, p_k), otherwise the path can be shortened. The shortest path from p^* to p_i can therefore be represented as $w(p^*, p_j) + d_G(p_j, p_i)$, for some node j that achieves the minimum in this sum. However, this is exactly the definition of $f_1(p_i)$. □

Corollary 7.2.3 *The values $f_1 = \{f_1(p_1) \ldots f_1(p_n)\}$ can be computed in time $O(n \log n)$ using Dijkstra's algorithm.*

Proof: By Theorem 7.2.2 the problem is of computing the shortest distances from node p^* to the rest of the nodes in the graph $\hat{G}$ defined above. The edge-weights of the graph are positive, and since the graph is planar $|\hat{E}| \leq 6|\hat{P}|$. Therefore, Dijkstra's algorithm on that graph will terminate in $O(n \log |(\hat{E})|) = O(n \log n)$. (See for example [24]). □

7.3 Selecting the new subset of nodes

After computing the new coarsening function values, the next step is that of forming the conflict-graph, and selecting a MIS of the conflict graph to be the new point set. The edge set of the conflict-graph CG(P) = (P, E) is defined as follows (see Definition 5.3.4):

$$E = \{(i,j) \mid \frac{f_1(x_i) + f_1(x_j)}{\beta} > \|x_i - x_j\|\}.$$

Or, when the graph metric is employed:

$$E = \{(i,j) \mid \frac{f_1(x_i) + f_1(x_j)}{\beta} > d_G(i,j)\}.$$

In the presence of boundaries the edges are also restricted to be in the domain (i.e. x_i and x_j are mutually visible). CG is given implicitly in terms of the underlying mesh, or Delaunay triangulation, G. Explicitly finding all the edges of CG could be expensive, since there could be $\Omega(n^2)$ edges.

It seems desirable to use the conflict graph information in an implicit manner only. A simple greedy algorithm to generate the MIS is to iteratively pick a new node from G, add it to the MIS, and remove from G all nodes that conflict with it. The edges of the conflict-graph are exposed only on a need-to-know basis. But how does one find all the nodes whose balls intersect with the selected node (i.e. share a conflict edge with it?). A natural solution is to search the graph G from the selected node. For this search procedure to be efficient, it is crucial to have a good criterion for deciding which nodes should be expanded in the search, and which should not. In particular, if there was a criterion such that only edges and nodes of G that have to be removed are expanded, the algorithm can be an $O(n)$ one, since a node or an edge is touched at most once; however, we currently have no such criterion.

The approach we suggest instead performs a search of the graph from the selected node p, expanding nodes as long as their graph distance from p is smaller than $3f_1(p)/\beta$. As the theorem below shows, if the function-balls of the two nodes intersect, they must be within that distance of each other. Note that this search procedure does not expose all the conflict graph edges since some nodes are invalidated before all of their conflict graph edges are touched.

This search procedure is efficient when coarsening by a small constant factor. The theorem below shows that in that case CG contains a linear number of edges, and therefore the search procedure perform only linear work. In the context of iterative coarsening schemes, like the one of Figure 7.2, each level is a coarsening by a constant factor of the previous level. Using the search procedure in these cases is therefore efficient.

Theorem 7.3.1 *Let f_0 be μ-Lipschitzian, and $M = (\mathcal{P}, E, \mathcal{B})$ a mesh such that $\mathcal{P}$ is β-spaced by f_0. Assume $\beta > 3$. Let $f_1 = f_{M,\mathcal{C}}$ be a coarsening of f_0 by a constant factor $\mathcal{C}$, using the graph metric. The following are true:*

1. *If node p_j is within a (graph metric) radius $3f_1(p_i)/\beta$ of node p_i, then*
$$f_0(p_j) \geq f_0(p_i)\frac{1}{\mathcal{C}}\left(1 - \frac{3}{\beta}\right).$$
2. *If the f_1-ball of point p_j intersects the f_1-ball of point p_i then*
$$\|p_i - p_j\| \leq 3f_1(p_i)/\beta.$$
3. *The ball of radius $3f_1(p_i)/\beta$ centered at node p_i contains at most $O(\mathcal{C}^4)$ nodes of $\mathcal{P}$.*

Proof:

1. By definition, $f_0 \leq f_1 \leq \mathcal{C} f_0$. Assume that node p_j is within radius $3f_1(p_i)/\beta$ of node p_i. Node p_j influenced the value $f_1(p_i)$ via the minimum operation, i.e.,
$$f_1(p_i) \leq \mathcal{C} f_0(p_j) + d_G(i,j) \leq \mathcal{C} f_0(p_j) + \frac{3f_1(p_i)}{\beta}$$
Hence,
$$f_0(p_i)(1 - \frac{3}{\beta}) \leq f_1(p_i)(1 - \frac{3}{\beta}) \leq \mathcal{C} f_0(p_j)$$
and the result follows.

2. By Lemma 6.3.1, if the two nodes have intersecting balls the distance between them is at most $2f_1(p_i)/(\beta-1)$. We search out for distance $3f_1(p_i)/\beta$, which is larger when $\beta > 3$.

3. We use a simple volume argument. The nodes of $\mathcal{P}$ are β-spaced by f_0, so balls of radius $f_0(p)/\beta$ around the nodes must be disjoint. Let node q be some node of $\mathcal{P}$ which is within distance $3f_1(p_i)/\beta$ from p_i. By part (1) of this proof,

$$f_0(q) \geq \frac{1}{\mathcal{C}}\left(1-\frac{3}{\beta}\right) f_0(p_i).$$

   Therefore, the balls of radius r around the nodes are disjoint, where

$$r = \frac{1}{\mathcal{C}}\left(1-\frac{3}{\beta}\right) \frac{f_0(p_i)}{\beta}.$$

   Since these balls are within distance $R = \frac{3\mathcal{C} f_0(p_i)}{\beta}$ from p_i, there can be at most $O(\mathcal{C}^4)$ of them.

□

Theorem 7.3.1 is important when opting for construction of the coarsening sequence from fine to coarse, with the mesh of level $i+1$ built from level i using a constant coarsening factor $\mathcal{C}$. Though this approach is theoretically vulnerable to the "repeated degradation" problem described in Chapter 5, it performs well in practice, as the experiments of Chapter 8 demonstrate. The next theorem shows this approach is also very efficient.

Theorem 7.3.2 *Let M_0 be a good aspect ratio mesh, and let f_0 be its spacing function. Let $M_1 \ldots M_1$ be the coarsening sequence generated using the algorithm of Figure 7.2. The cost of creating that sequence is $O(\Sigma_{i=0}^{i=k-1}|M_i|\log|M_i|)$.*

Proof: The algorithm of Figure 7.2 generates mesh M_{i+1} by calling the algorithm of Figure 7.3 at each level. Step (1) of that algorithm, computing the coarsening function over the mesh M_i, takes $O(|M_i|\log|M_i|)$ using the approach of Section 7.2. The subroutine used is a simple Dijkstra's shortest paths subroutine. Step (2) coarsens the boundary. The boundary coarsening method suggested in the next section costs $O(n \log n)$, where n is the number of boundary nodes. Step (3) is obviously linear. In step (4), by Theorem 7.3.1, each node of M_i searches a constant number of neighboring nodes. The final Delaunay triangulation of step (5) again costs $O(|M_i|\log|M_i|)$. □

The algorithm description so far omitted treatment of the boundaries. The next session discusses the issues of more general boundary coarsening.

7.4 Treatment of the boundaries

Perhaps the greatest challenge is that of dealing with complicated boundaries appropriately. The coarsening algorithm presented in Chapter 5 used a somewhat simplified framework. Recall the mesh $M = (\mathcal{P}, E, \mathcal{B})$. $\mathcal{B}$ is the boundary description, a set of segment that may intersect only at their end-points. M discretizes the boundary and thus may place extra nodes on the boundary segments. To distinguish the two, we refer to the boundary segments end-points, and the boundary points, as the *boundary vertices* whereas the set $\mathcal{P}$ is the set of mesh nodes. Let S be the set of boundary vertices. Let Q be the set of mesh nodes placed on the boundary, i.e., $Q = \mathcal{P} \cap \mathcal{B}$. Clearly, $S \subseteq Q$. The simplified framework of Chapter 5 assumed that all the meshes of the coarsening

Procedure: MULTI_LEVEL_GRAPH_METRIC_COARSEN(M_0, f_0, β)

1. Let k be the length of the required coarsening sequence:
2. For $i = 1$ to k

   $(M_i, f_i) =$ GRAPH_METRIC_COARSEN$(M_{i-1}, f_{i-1}, 2, \beta)$
3. Return $(M_1, ..., M_k)$.

Figure 7.2: Multi level function-based coarsening using the graph metric.

Procedure: GRAPH_METRIC_COARSEN$(M_0, f_0, \mathcal{C}, \beta)$

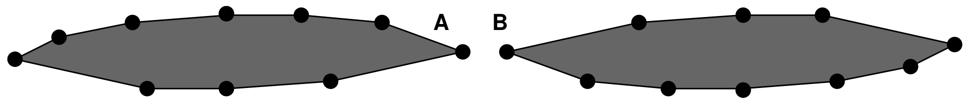
Input: $M_0 = (\mathcal{P}_0, E_0, \mathcal{B}_0)$, a mesh over a domain described by $\mathcal{B}_0$
f_0, a spacing function defined on the nodes $\mathcal{P}_0$
β, the spacing constant
$\mathcal{C}$, the coarsening factor

Output: a mesh $M_1 = (\mathcal{P}_1, E_1, \mathcal{B}_1)$, and a spacing function f_1 on the nodes $\mathcal{P}_1$.

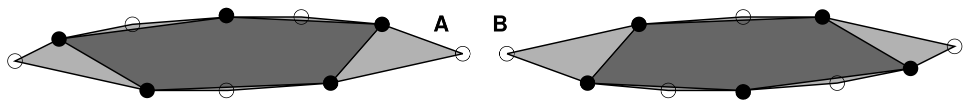
Method:

1. Compute $f_1(\mathcal{P}_0)$ using the algorithm of section 7.2.
2. Let $(\mathcal{P}_1, V, \mathcal{B}_1) =$ BOUNDARY_COARSEN. $\mathcal{P}_1$ is the iteratively constructed node set. After this call it contains the new boundary nodes. V is a vector of labels marking which nodes have been invalidated by the boundary coarsening. $\mathcal{B}_1$ is the new boundary.
3. Let Π be some ordering on the nodes $\mathcal{P}_0$. This ordering can be either randomized or some priority assignment to the nodes.
4. For $i = 1$ to $|\mathcal{P}_0|$
   (a) let $n = \Pi(i)$ be the current node. If $n \in \mathcal{P}_1$ already or $V(n) =$invalid, skip rest of loop body.
   (b) add the new node in: $\mathcal{P}_1 = \mathcal{P}_1 \cup \{n\}$.
   (c) using the graph structure M_0, search from the node n using Dijkstra's algorithm (closest node first) to find all graph nodes k such that $d_G(n, k) \leq 3f_1(n)/\beta$.
      - if $f_1(k) + f_1(n) > \beta d_G(k, n)$, set $V(k) =$invalid.
5. Return $M_1 = (\mathcal{P}_1, \mathrm{DT}(\mathcal{P}_1, \mathcal{B}_1), \mathcal{B}_1)$ and f_1 restricted to $\mathcal{P}_1$.

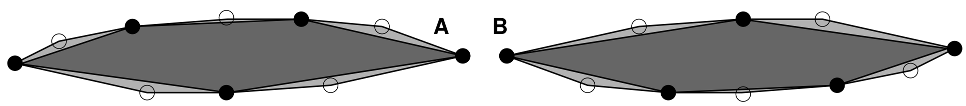
Figure 7.3: one level function-based coarsening using the graph metric.



Original boundary contains two holes. Two boundary points on the holes, A and B, are very close to each other.



Because nodes A and B were too close in terms of the features around them, they must be coarsened to enable a good aspect ratio triangulation of the interior.



The best approximation to the boundary retains point A and B. Since A and B are too close, they will force the appearance of a bad aspect ratio triangle in any triangulation of the interior.

Figure 7.4: Good aspect ratio versus approximation boundary coarsening.

sequence use the same boundary description $\mathcal{B}$, but that nodes of Q may be coarsened out. This is achieved by coarsening only as long as $f_1 < \mathsf{lfs}_{\mathcal{B}}$ where f_1 is the coarsening function and $\mathsf{lfs}_{\mathcal{B}}$ is the local feature size induced by the boundary $\mathcal{B}$. (See Definition 2.2.5). It is easy to show that S is always retained under this restriction.

This section deals with the issues arising when attempting to coarsen beyond this restriction, and when the boundary itself may be coarsened. The new boundaries $\mathcal{B}_1$ of the coarser mesh $M_1 = (\mathcal{P}_1, E_1, \mathcal{B}_1)$ must be formed, and can be different from $\mathcal{B}$.

Boundary coarsening is such a challenge since two goals or constraints have to be addressed at the same time: (1) coarsening to produce a good aspect ratio mesh (2) coarsening to approximate the original boundaries.

Both requirement stem from the subsequent numerical method needs. In order for the numerical solution to converge to the unknown function u of the PDE, it suffices that the mesh is triangulated using good aspect ratio triangles, see for example [1].

In order for the numerical boundary condition to approximate the real boundary condition, the discretized boundary should approximate the continuous boundary. Meaning, the error between the coarse discretized boundary and the continuous boundary should be small under some measure: e.g. the volume difference, or absolute distance.

Figure 7.4 gives a case where the best boundary approximation coarsening runs contrary to a good aspect ratio coarsening, and forces the appearances of small angles in the triangulation. Note, however, that a good aspect ratio coarsening approximates the boundaries well with an error proportional to the local edge-length; the problem is that a much better boundary approximation, using a similar number of boundary points, exists. Which requirement is more important, good aspect ratio or boundary approximation, is a decision to be made based on the numerical method

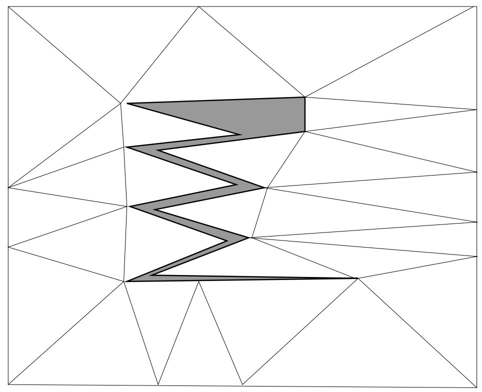


Figure 7.5: Boundary nodes can be necessary for maintaining boundary shape. Coarsening beyond this point can cause a coarse boundary that intersects itself, and totally unlike the current boundary.

needs.

The problem of approximating the boundary is well-researched, and is usually referred to as "boundary simplification." Garland and Heckbert [36] have an excellent survey of this area. Since boundary simplification does not guarantee a good aspect ratio coarsening, this section continues with a discussion of function-based coarsening of the boundaries, a method that attempts to provide such guarantees. We concentrate on the case where a mesh with its spacing function is presented, the coarsened spacing function has been computed, and now the boundary has to be coarsened to accommodate the coarse spacing function.

7.4.1 Issues of function-based boundary coarsening

As before, let $\mathcal{B}$ be the initial boundary, and let Q be the set of mesh nodes placed on the boundary. S, the set of boundary vertices, plays no part in the simple algorithm discussed here.

The coarsening spacing function f_1 controls the spacing of the final mesh nodes, and of the final mesh boundary nodes. The methods presented in Chapter 5 can be applied to the boundary coarsening as well. The new boundary, $\mathcal{B}_1$, is constructed in two phases. The first phase generates the new set of boundary nodes, Q_1, by function-based coarsening the boundary nodes Q. That is, a conflict-graph is created over the nodes Q, and the set Q_1 is selected as a maximal independent set of the conflict graph. The second phase constructs the new boundary $\mathcal{B}_1$, using node contraction. When a node q of Q is eliminated from being selected into the new set of boundary nodes during the maximal independent set, one of its boundary neighbors that has not been eliminated yet is chosen. Call this neighbor p. q is then contracted to p. All the boundary edges incident to q are thus transformed to boundary edges incident to p. Non-boundary edges out of q are not contracted.

This simple method encounters the following difficulties:

boundary integrity: The biggest problem is that this procedure does not guarantee geometric correctness of $\mathcal{B}_1$, in particular the new boundary might intersect itself. This can happen even if $\mathcal{B}$ is topologically a very simple closed loop, see Figure 7.5. Furthermore, there are instances where all possible selections of a maximal independent set lead to intersecting boundary, as Figure 7.6 shows. Nodes a and b conflict, therefore either none or one of them have to be selected. If none are selected,

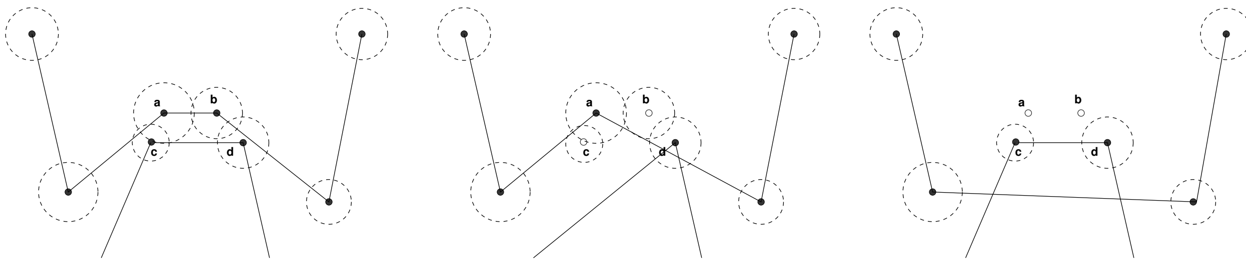


Figure 7.6: An example of a boundary, with a coarsening function, such that all possible selections of a MIS of the conflict-graph lead to self-intersecting boundaries. (a) the original boundary; coarsening function values denoted by the radius of the dashed balls around nodes. (b) one possible selection of the MIS, with node a selected. (c) the shape of the boundary when neither a nor b are selected. The remaining possibility is selecting b, and is symmetric to (b).

Figure 7.6(c) shows that the resulting boundaries intersect. If only one is selected, we assume without loss of generality that it is node a, Figure 7.6(b) shows that the resulting boundaries intersect as well.

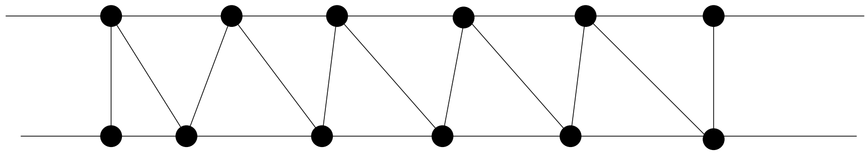
This problem may occur whenever $f_1 > \frac{\beta}{2}\mathsf{lfs}_{\mathcal{B}}$, since at that stage the f_1-balls around the boundary nodes start to intersect. Therefore, integrity of the boundary shape can be guaranteed only as long as $f_1 < \frac{\beta}{2}\mathsf{lfs}_{\mathcal{B}}$. In practice, it is desirable to continue beyond that point. The method should be combined in that case with a routine to detect intersection of the boundary. (e.g. the simple sweep-line method to detect intersection at cost $O(n \log n + k)$, where n is the number of edges and k is the number of intersections. Note the routine can be stopped when an intersection is detected, for a total cost of detection $O(n \log n)$).

For the rest of the discussion, we assume that the new coarse boundary $\mathcal{B}_1$ does not intersect itself. Even after passing this hurdle, good aspect ratio is guaranteed only if f_1 is similar to or smaller than $\mathsf{lfs}_{\mathcal{B}_1}$. Note, that this time $\mathsf{lfs}_{\mathcal{B}_1}$ is used rather than $\mathsf{lfs}_{\mathcal{B}}$: the local feature size of the **coarsened** boundary rather than of the original boundary $\mathcal{B}$. The following paragraph discusses that:

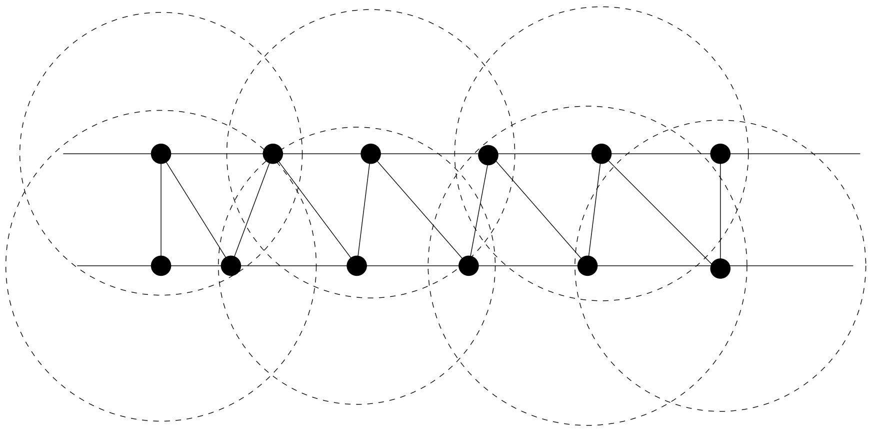
good aspect ratio guarantees: after coarsening, the point set Q_1 is spaced according to f_1. The new boundary $\mathcal{B}_1$ is a graph over Q_1. However, it does not necessarily follow that $\mathcal{B}_1$ accommodates f_1. In particular, it could be the case that $\mathsf{lfs}_{\mathcal{B}_1} \ll f_1$. For a simple example of this occurring, see Figure 7.7. The nodes are spaced according to f_1 along the edges of a long pipe, but the lfs is dictated by the distance between the two long pipe edges. The intuition behind this example is that whereas the nodes could be coarsened to accommodate f_1, the shape of the domain can not be coarsened.

The problem with $\mathsf{lfs}_{\mathcal{B}_1} \ll f_1$ is that, as Ruppert [64] has shown, any good aspect ratio mesh must be spaced according to the lfs of its boundary, hence no good aspect ratio triangulation is possible when $\mathsf{lfs}_{\mathcal{B}_1} \ll f_1$. As a conclusion, good aspect ratio coarsening at this stage can be guaranteed only as long as f_1 is similar to or smaller than $\mathsf{lfs}_{\mathcal{B}_1}$.

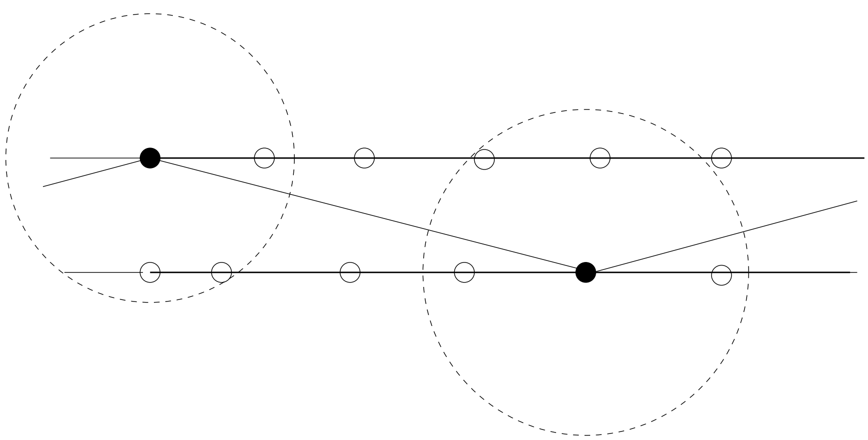
Now we assume the last two problems identified do not arise, i.e. $\mathcal{B}_1$ does not intersect itself and $f_1 < \mathsf{lfs}_{\mathcal{B}_1}$. The following precaution has to be taken so that triangles of bad aspect ratio do not form near the coarsened boundaries:



A simple mesh, inside a long pipe.



The coarsening function is indicated by the radius of the dashed lines. Note that the local feature size is ruled by the distance between the pipe's two edges.



The new node set, spaced according to the coarse spacing function. Note that the local feature size did not change, and it is still ruled by the distance between the two pipe edges. In this case, the result of the discrepancy between the local feature size and the coarse spacing function is a triangulation with small angles.

Figure 7.7: When the local feature size of the coarsened boundary is small compared to f, even though the boundary points are spaced according to f, small angles can occur.

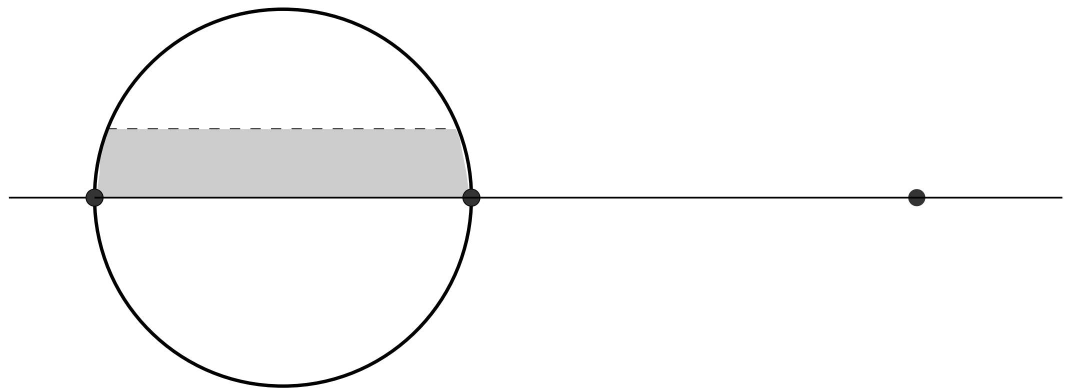


Figure 7.8: To ensure interior nodes do not get too close to boundary edges, a protecting zone around each edge is maintained. Interior nodes in the protective zone are disqualified. The protecting zone is the intersection of the diametrical ball of the edge and a rectangular zone. The hight of the rectangle is $\mathcal{R}$ times the edge length. $\mathcal{R}$ is an input parameter.

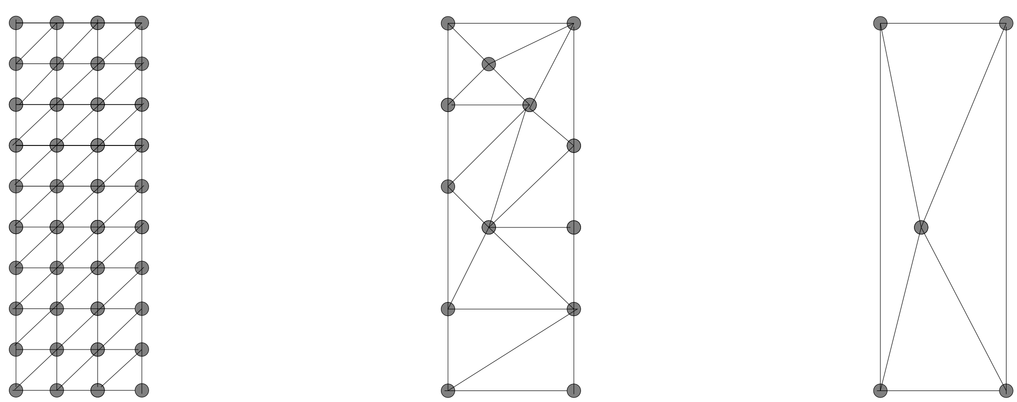


Figure 7.9: If the boundaries are not protected, an interior node can get closer and closer to the boundary relative to the spacing function.

Protecting the coarsened boundary: It is important to add a mechanism that prevents interior nodes from getting too close to the boundary, otherwise flat triangles of bad aspect ratio can form.

One possible solution is to disallow interior nodes within the diametrical circle of an edge, a circle centered on the edge and touching both the nodes incident on the edge. This results in a conforming Delaunay triangulation, but has the tendency of introducing longer, thiner triangles near the boundaries.

Instead, we introduce a ratio parameter $\mathcal{R}$. All nodes falling within the diametrical circle of a new edge are tested using the following:

$$\frac{\text{distance}(node, edge)}{\text{edge length}} \geq \mathcal{R}$$

and a node is disqualified if it is too close. Look at Figure 7.8, the disqualifying area is shaded in the figure. In the implementation, we set $\mathcal{R} = 0.2$ for function-based coarsening, and $\mathcal{R} = 0.15$ for MIS-based coarsening. The later is set lower since MIS-based coarsening tends to coarsen the boundary much more quickly, and the MIS results were better when less nodes were disqualified by the boundaries.

7.5 Summary

This chapter presented a practical algorithm for mesh coarsening that is based on the theoretical function-based coarsening framework developed in Chapter 5.

The practical algorithm coarsens the graph iteratively, thus trading off efficiency and simplicity versus the theoretical guarantees against repeated degradation. The experiments of the next chapter demonstrate that this trade-off has no drawbacks in practice and that the resulting coarsening sequences are of high-quality even on complex domains and highly graded and unstructured meshes without showing signs of repeated degradation.

Furthermore, the practical algorithm uses the weighted graph metric instead of the Euclidean metric. The two metrics are similar over Delaunay triangulations. The weighted graph metric has two crucial advantages: (1) it easily handles holes and obstacles in the domain (2) coarsening functions computations can be carried out efficiently using a shortest paths algorithm.

Overall, these modifications lead to a simple and efficient algorithm that is very easy to implement and generates the mesh M_i of the coarsening sequence at cost $O(n_i \log n_i)$.

Procedure: BOUNDARY_COARSEN($M, f, \beta, \mathcal{R}, F$)

Input: $M = (\mathcal{P}, E, \mathcal{B})$, a mesh
f, β: spacing function and its spacing constant
$\mathcal{R}$: edge protection ratio
F: set of nodes to force into the output node set

Output: $\mathcal{P}_1$: output boundary nodes
V: valid/invalid marks for the node set $\mathcal{P}$
$\mathcal{B}_1$: new boundary edges

Method:

1. let $Q = \mathcal{P} \cap \mathcal{B}$ be the boundary nodes.
   let G be the boundary edges.
   let $V(i)$=valid for all nodes $p_i \in \mathcal{P}$
   let $\mathcal{P}_1 = F$

   for $i = 1$ to F

   Using the graph structure of M, search from the node i using Dijkstra's algorithm (closest node first) to find all graph nodes k such that $d_G(i, k) \leq 3 f_1(i)/\beta$.
   if $f(k) + f(i) > \beta d_G(k, i)$, and $k \notin F$, set $V(k)$ =invalid.

   end if

   let Π be some ordering on the nodes Q.

2. for $i = 1$ to $|Q|$

   (a) let $n = \Pi(i)$ be the current node. If $n \in \mathcal{P}_1$ already or $V(n)$ =invalid, skip rest of loop body.
   (b) set $\mathcal{P}_1 = \mathcal{P}_1 \cup \{n\}$.
   (c) Using the graph structure of M, search from the node n using Dijkstra's algorithm (closest node first) to find all graph nodes k such that $d_G(n, k) \leq 3 f_1(n)/\beta$.
      - if $f(k) + f(n) > \beta d_G(k, n)$, set $V(k)$ =invalid.

   end if

3. generate the edge set of $\mathcal{B}_1$ by node contraction.

4. edge protection: let $e \in \mathcal{B}$ be a new edge. Assign $V(p)$=invalid to every interior node $p \in \mathcal{P} \setminus \mathcal{P}_1$ in the diametrical circle of e, whose distance d from e is too small, i.e., $d < \mathcal{R}|e|$. These nodes are found by a graph search from the vertices of e.

5. Return $(\mathcal{P}_1, V, \mathcal{B}_1)$.

Figure 7.10: Basic boundary function-based coarsening procedure. This procedure allows the user to input a list of special points F are forced into the set of new boundary nodes. This is very useful to ensure, for example, that the four vertices of a square domain are not coarsened out. In the experiments, this option has been implemented and used to guarantee that closed curves of the boundary always contain three points. For other boundary coarsening strategies, see the related work in Section 5.2.1.

Chapter 8

Coarsening - Experimental Results

The most popular approaches to mesh coarsening are topological, such as the Maximal Independent Set (MIS) method discussed in Section 5.2.1. The drawbacks of this approach, which were pointed out in the previous chapters, might seem to be oddly contrasted with its popularity. This chapter partially explain this popularity by showing that in practice, these problems are not severe when coarsening quasi-uniform meshes. However, for more graded meshes, topological approaches fail theoretically as well as experimentally. Therefore, the function-based coarsening approach introduced in this thesis is necessary to obtain a good-quality coarsening sequence.

We implemented two algorithms: an MIS-based algorithm, and the practical function-based approach developed in Chapter 7. Both algorithms are fairly simple and efficient. The goal of this chapter is to compare and contrast the performance of these algorithms over two types of meshes: quasi-uniform meshes and graded unstructured meshes. The comparison is in terms of the quality of the coarsening sequences the algorithms produce. The experiments of this chapter show that for graded mesh the function-based approach outperforms the MIS-based approach in practice as well as in theory.

Section 8.1 presents the collection of meshes over which both the MIS-based method and the function-based method are measured. Section 8.2 discusses the implementations. Section 8.3 discusses the affect the different function-based coarsening parameters have on the quality of the resulting coarsening sequence, and fixes the parameter setting for the latter experiments. Section 8.5 presents the results for MIS-based and function-based coarsening over grid-like meshes, and Section 8.6 shows the results for graded meshes.

8.1 Test suite

Examples for 2d graded meshes are not abundant in the public domain. The NASgraph [23] collection is the only source we found containing such examples. The four meshes constituting our test suite are available off the NASgraph homepage, which contains many more contributed examples of matrices, graphs and meshes (unfortunately most of them are not appropriate for our experiments).

The "crack plate" mesh The "crack plate" mesh was generated by Omar Ghattas and Xiaogang Li of Carnegie Mellon University. The physical problem modeled by the mesh is a plate with a horizontal crack running from the middle of the left edge to the center of the plate [31]. See Figure 8.1.

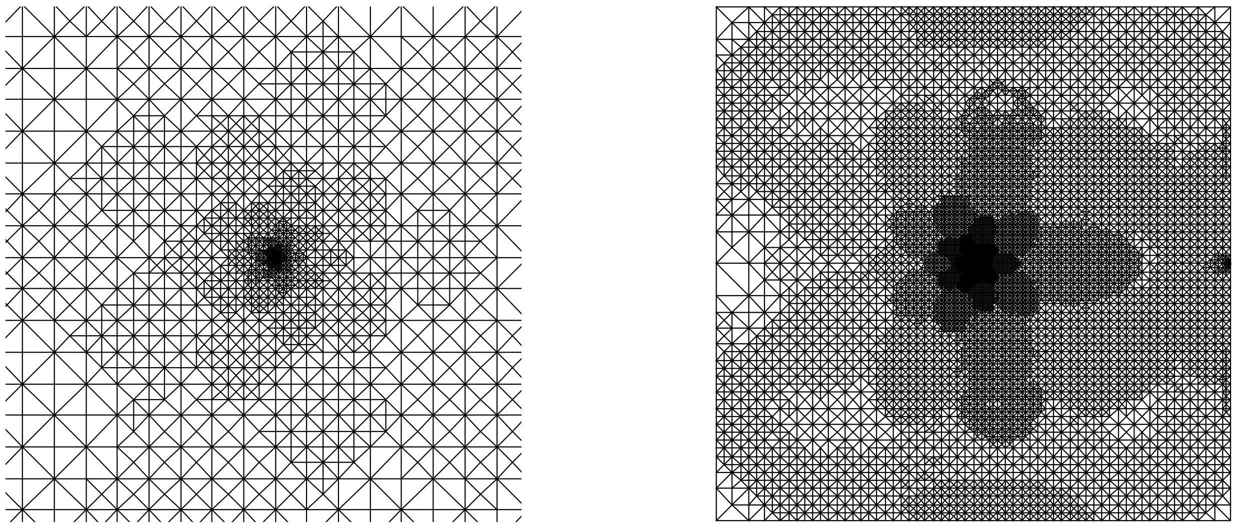


Figure 8.1: crack plate mesh: on the left, a zoom into the center; on the right, the mesh.

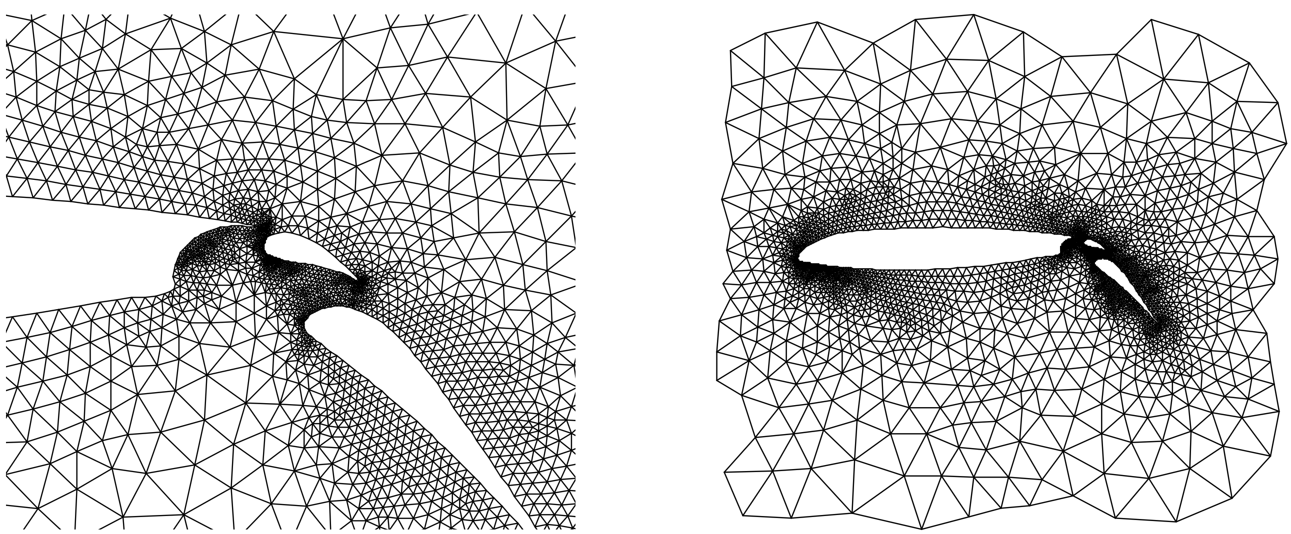


Figure 8.2: AIRFOIL 1 mesh: on the left, a zoom onto the wing; on the right, the mesh.

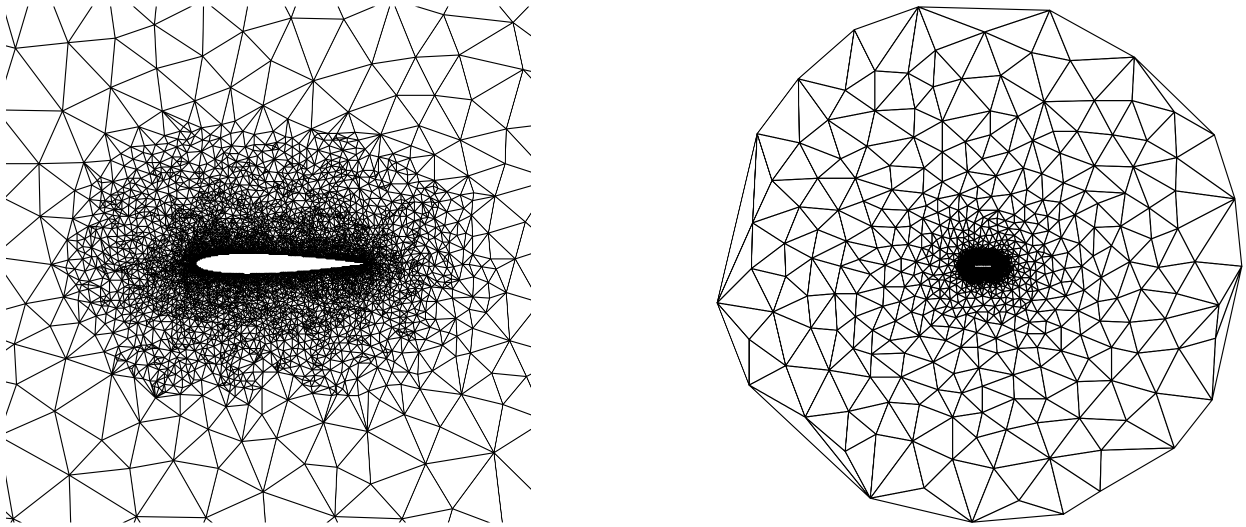


Figure 8.3: AIRFOIL 2 mesh: on the left, a zoom onto the wing; on the right, the mesh.

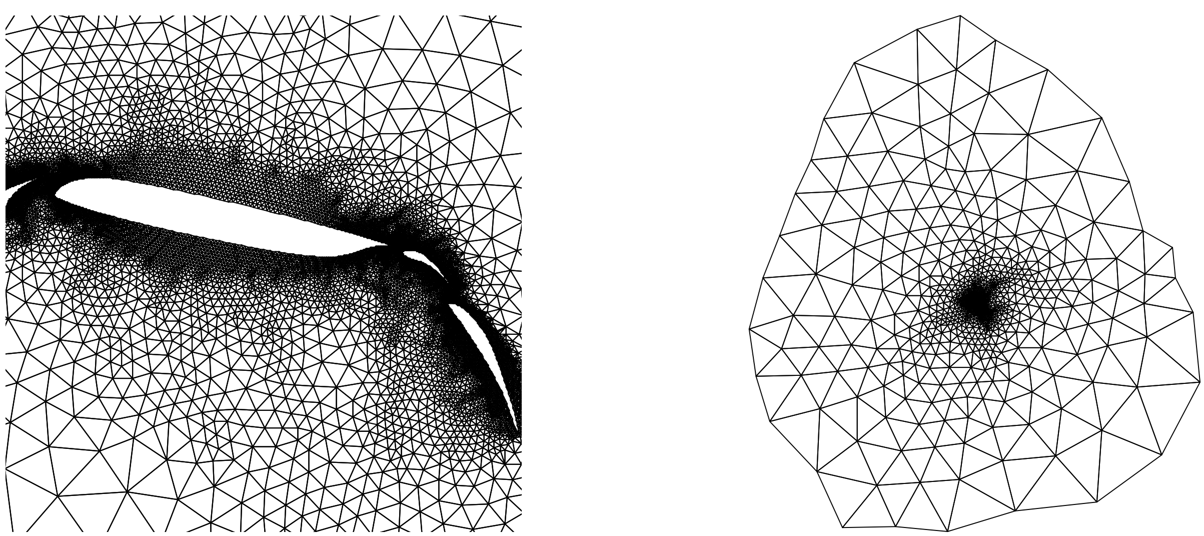


Figure 8.4: AIRFOIL 3 mesh: on the left, a zoom onto the wing; on the right, the mesh.

The "air foil" meshes The physical problem modeled by the air foil meshes is the simulation of turbulence over a wing. The meshes model a cross section of the wing. We refer to them as AIRFOIL 1, AIRFOIL 2, and AIRFOIL 3. Whereas the NASgraph collection does not supply references to the person generating the mesh, we believe that at least the mesh AIRFOIL 1 was generated by Barth and Jespersen. See Figures 8.2, 8.3, and 8.4.

The "grid" meshes We use the term "grid" meshes to refer to regular triangular meshes whose nodes are laid in a grid. This mesh is very uniform, and is included in the test suite to compare the behavior of the MIS-based coarsening and the function-based coarsening for the case of uniform and quasi-uniform meshes.

The grid meshes all have million nodes, and are parameterized by their aspect ratio, i.e. the mesh grid(a) is composed of nodes placed at coordinates (i, aj) where $1 \leq i, j \leq 1,000$.

Mesh	Size		Gradation		Angles			
	nodes	triangles	g= $\log \frac{E}{e}$	$\frac{\lvert\text{triangles}\rvert}{\text{g}}$	smallest	largest	number $\leq 10°$	number $\leq 20°$
CRACK	10183	20141	15.43	1305	37.00°	97.61°	0	0
AIRFOIL 1	4253	8034	9.55	841	8.18°	117.08°	1	7
AIRFOIL 2	6691	13057	12.93	1009	7.11°	163.94°	9	284
AIRFOIL 3	15606	30269	14.89	2032	9.39°	138.48°	1	24
GRID(a)	1000000	1996002	$\log a$	$\frac{1996002}{\log a}$	$\arctan \frac{1}{a}$	90°	depends on a	

Table 8.1: Properties of the test suite meshes

Remarks: Table 8.1 lists properties of the test suite meshes such as their size, angle quality, and gradation. How graded the mesh is can be measured by taking the log of the longest edge over the shortest edge. This indicates how many orders of magnitudes are spanned by the edge lengths. Further more, dividing the number of triangles by the number of scales, shows how many triangles are on average at every scale level. The mesh CRACK is the most graded mesh, especially when compared with the mesh closest to it in gradation, AIRFOIL 3, it clearly has far less edges at each scale level. Mesh CRACK, however, has very simple boundaries, so the boundary coarsening strategy does not cloud the performance of the function-based coarsening and in that sense the mesh is easier to coarsen.

8.2 Implementations

The two algorithms we experimented with are the MIS-based coarsening, and the function-based coarsening, both using our own implementation of the algorithms.
Our implementations use the following general framework:

1. The input includes the initial mesh $M_0 = (\mathcal{P}_0, \mathcal{T}_0, \mathcal{B}_0)$, described by:
    - A set of points $\mathcal{P}_0$

- A set of triangles over the points $\mathcal{P}_0$, describing the triangulation.
- A description of the boundary of the mesh: a set of edges of $\mathcal{B}_0$ over the nodes $\mathcal{P}_0$. A node that is incident to an edge of $\mathcal{B}$ is called a boundary node.

The input further contains a set of special points F. These points may not be eliminated during the coarsening process. Over the test-suite meshes, the following special points are used:

- CRACK mesh: four end points of the enclosing square. This enables the coarsening to always produce a square domain.
- AIRFOIL 1, AIRFOIL 2, and AIRFOIL 3: From the enclosing boundary, three or four points to ensure the outer boundary shape integrity. Furthermore, each hole has three special points. This enables coarsening not to degenerate holes beyond triangles. This condition can be enforced automatically in a more sophisticated implementation.

2. Coarsening proceeds iteratively. At each level, the mesh is built from the previous level's mesh information only.

3. At each level, the new mesh M_{i+1} is generated from mesh M_i in increasing order of dimension of the mesh parts: first the special points, then the new boundary edges, and finally the interior points. Let $\mathcal{P}_{i+1}$ and $\mathcal{B}_{i+1}$ stand for the new mesh's point set and boundary edges.

   - $\mathcal{P}_{i+1}$ is initialized to F, the set of special points, marking the appropriate points of $\mathcal{P}_i$ as invalid for addition in the following stages.
   - A subset of the boundary points of $\mathcal{B}_i$ is selected for addition to $\mathcal{P}_{i+1}$, marking the appropriate points of $\mathcal{P}_i$ as invalid for selection latter. The points of $\mathcal{B}_i$ are considered for addition in a randomized order. The rule for selection depends on the method (MIS or function-based). The new boundary is constructed from $\mathcal{P}_{i+1}$ by node contraction of the eliminated boundary nodes from $\mathcal{P}_i$.
   - Points of $\mathcal{P}_i$ that are too close to the new boundary edges are marked as invalid for addition into $\mathcal{P}_{i+1}$ in the next stage. A ratio parameter $\mathcal{R}$ is used. A point p is too close to an edge if it is within the diametrical circle of the edge *and* its distance to the edge is smaller than $\mathcal{R}$ times the edge length.
   - The valid points of $\mathcal{P}_i$ are considered for addition into $\mathcal{P}_{i+1}$ in a randomized order. The rule for node selection depends on the method (MIS or function-based).

4. Finally, the mesh M_{i+1} is a constraint Delaunay triangulation of $\mathcal{P}_{i+1}$ with the constraints $\mathcal{B}_{i+1}$.

The rule for selection of nodes in the MIS-based method is that a point must have no edge of $\mathcal{M}_i$ incident to previously chosen points. For more details on the MIS algorithm, see Section 5.2.1. The rule for node selection in the function-based method is that the balls around the points, whose radiuses are proportional to the function size, do not intersect balls of previously selected nodes.

The ratio parameter $\mathcal{R}$ is set to be 0.2 for function-based coarsening, and 0.15 for MIS-based coarsening. $\mathcal{R}$ is set lower for the MIS-based method since the boundary edges in this case grow quicker, and otherwise too many internal points were pruned out, adversely affecting the mesh quality. Note that some protection of the edges is necessary, otherwise bad triangles form on the boundary edges. The aspect ratio of a triangle on the boundary is proportional to the ratio parameter $\mathcal{R}$.

The parameters of the function based coarsening, the spacing constant β and the coarsening factor $\mathcal{C}$, are set differently for uniform and graded meshes. $\beta = 25$ in both cases, but $\mathcal{C} = 2$ in the uniform case and $\mathcal{C} = 8$ in the graded case. The following section describes how the settings were picked for the graded meshes. For the uniform case, $\mathcal{C} = 8$ does produce a good coarsening, but the size reduction is larger than the MIS case. To enable the comparison, a smaller $\mathcal{C}$ was used.

8.3 Function-based coarsening parameters

Two parameters of the function based algorithm can be set during run-time invocation at the user's discretion. These are the coarsening factor $\mathcal{C}$, determining the relative coarsening between subsequent meshes, and the spacing constant β, controlling the smoothness of the coarsening functions.
Recall the phases of coarsening:

- Initially, a spacing function is recovered from the original mesh $M_0 = (\mathcal{P}_0, \mathcal{T}_0, \mathcal{B}_0)$. The function used in our implementation is:

$$f_0(p) = 0.5 \min_{(p,q)\in E} \|p - q\|$$

- The coarsening function f_{i+1}, defined over P_i, is:

$$f_{i+1}(p) = \min_{q\in P_i} \mathcal{C} f_i(q) + d_{\mathcal{M}_i}(p, q)$$

  Where $d_{\mathcal{M}_i}$ is the weighted graph-metric distance between q and p, using mesh M_i as the graph. (See Definition 7.1.2).

- The new point set P_{i+1} is a maximal subset of P_i, such that balls of radius f_{i+1}/β do not intersect:

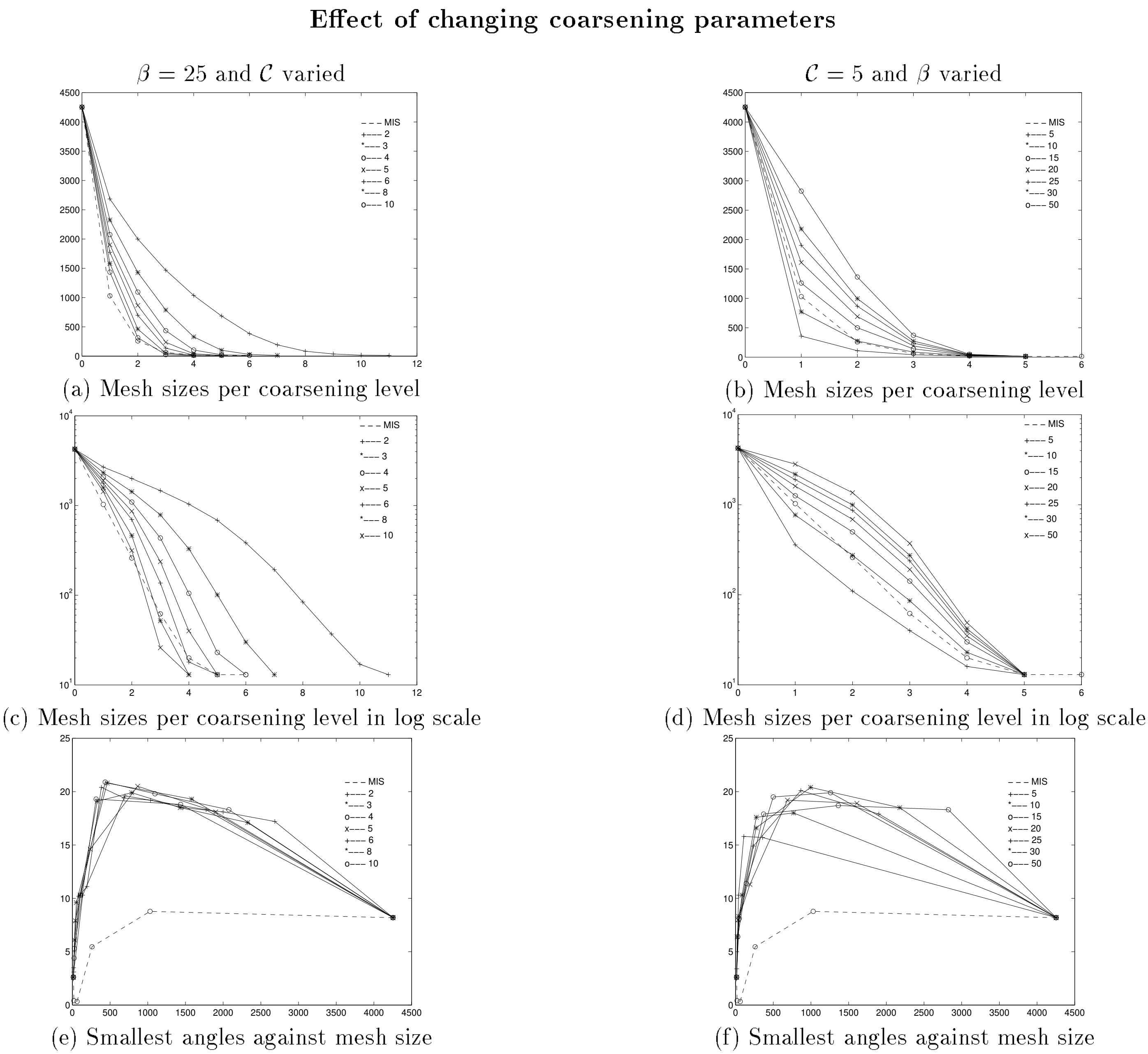
$$\forall p, q \in P_{i+1} \quad f_{i+1}(p) + f_{i+1}(q) \leq \beta d_{\mathcal{M}_i}(p, q)$$

Clearly, β and $\mathcal{C}$ interact. For example, assume $f_0 = s$, some constant. Then, $f_i \approx \mathcal{C}^i s$, and the radius of the balls around mesh nodes is $\mathcal{C}^i s/\beta$ .To somewhat offset that, the first level coarsening factor is actually set to $\mathcal{C}\beta$. Subsequent coarsening factors used are $\mathcal{C}$. For constant functions, this sets the spacing around points to the correct value of $\mathcal{C}^i s$.

To gauge the way $\mathcal{C}$ and β effect the mesh coarsening sequence properties, we ran a small experiment on the mesh AIRFOIL 1. First, the spacing constant β is fixed to the value 25, whereas $\mathcal{C}$ is varied over the set $\{2, 3, 4, 5, 6, 8, 10\}$. Then, $\mathcal{C}$ is fixed to be five, and β is varied over the set $\{5, 10, 15, 20, 25, 30, 50\}$. This strategy is used both to avoid running over all 49 combinations, and to isolate the effects of each parameter.

The resulting qualities of the function-based coarsening sequence are compared with the corresponding qualities of the MIS-based coarsening of the same mesh. See Figure 8.5.

The mesh sizes (Figure 8.5[a-d]): Five experiments in each parameter setting were performed. Each experiment results in a coarsening sequence. The figure plots the average size of the mesh against the level (location) in the sequence, in regular coordinates (sub-figures (a) and (b)) and semi-log scale (sub-figures (c) and (d)). As is to be expected, when the coarsening factor is increased while keeping β fixed, the meshes are coarsened faster. When the coarsening factor is eight and larger, the coarsening rate seems to equate that of the MIS-based coarsening. Clearly, however, all the setting



(a) Mesh sizes per coarsening level

(b) Mesh sizes per coarsening level

(c) Mesh sizes per coarsening level in log scale

(d) Mesh sizes per coarsening level in log scale

(e) Smallest angles against mesh size

(f) Smallest angles against mesh size

Figure 8.5: Effect of changing coarsening parameters on the coarsening sequence size and aspect ratio quality.

of the coarsening factor offer at least a geometrical rate of size reduction, though the ratio between the sizes of subsequent meshes in the sequence is smaller for smaller settings of $\mathcal{C}$. Increasing β has similar effect to decreasing $\mathcal{C}$. For faster size reduction, a higher setting of $\mathcal{C}$ and a lower setting of β both seem desirable.

The mesh element shape (Figure 8.5[e-f]): Five experiments in each parameter setting were performed. Each experiment results in a coarsening sequence. The figure plots the average smallest angle of the meshes against the average size of the mesh at this level. Both averages are taken over the five experiments. Looking at Figure 8.5(e), it seems changing the coarsening factor does not affect the angle quality of the coarsening sequence, and the curves closely coincide. Notice that the angle quality curve for function-based coarsening is superior to that of the MIS-based method.

Looking at Figure 8.5(f) it is clear that decreasing β adversely affects the angle quality. β should therefore be kept at a reasonably high value, lest the angle quality is compromised. Notice however that even the worst settings of β lead to a better angle-curve than the MIS method.

To summarize, both β and $\mathcal{C}$ can increase the size reduction rate, but the settings of β that increase the reduction rate badly affect the angle quality; β should be tuned for angle quality, and $\mathcal{C}$ for size reduction. What prevents us from increasing $\mathcal{C}$ infinitely is that subsequent meshes should have bounded intersection, i.e. an element of mesh $\mathcal{M}_{i+1}$ should intersect a small number of elements of $\mathcal{M}_i$. In the experiments the settings we used were $\mathcal{C} = 8$ and $\beta = 25$.

8.4 Measured depth and work constants

The algorithm implemented constructs the mesh M_{i+1} from the mesh M_i at cost $O(n_i \log n_i)$, n_i being the size of M_i. This section provides estimates of the constants in the basic phases of the algorithm.

8.4.1 Computing the coarsening function

The coarsening function is computed using a simple shortest-paths subroutine as described in Section 7.2. The shortest-paths routine implicitly constructs a spanning tree of the graph, where the parent of a node i is a node j whose function value is used to fix node i's function value. Whereas evaluating the work constants of this standard algorithm is of no great interest, the depth of this spanning tree is highly dependent on the coarsening problem, and is of interest for future attempts to parallelize the algorithm.

Therefore, experiments were run measuring the maximal depth of the spanning tree and also the average depth of a node in a tree.

Ten tests each were run for each of the four meshes and two coarsening factors: $\mathcal{C} = \{4, 8\}$, for a total of eighty experiments. The spacing constant β is fixed at $\beta = 25$ during the experiments.

The table below reports the results. Each row of the table corresponds to a level in the coarsening sequence. The *max* column corresponds to the maximal depth of the spanning tree over the ten experiments, whereas the *avg* column corresponds to taking the maximum over the ten experiments of the average depth of a node in the spanning tree.

coarsening level:	initial mesh is at level 0, subsequent coarsenings at higher levels.
number of nodes:	an average, over the experiments, of the number of nodes of the coarsening sequence mesh at that level.
angles: **[non] boundary** **smallest**	the smallest angle observed at that coarsening level over the set of experiments. For non-boundary, only triangles that have no boundary nodes are considered. For boundary, only triangles that have a boundary node are considered.
angles: [non] boundary **avg small.**	the average of the smallest angles. Each experiments contributes the smallest angle of the coarsening sequence at that level, an average over the experiments is taken. at that coarsening level over the set of experiments.
angles: ≤ 10, **non-boundary**	the total number of small angles observed at that level, over all the experiments, that originated from triangles with no boundary node, divided by the number of experiments.
angles: ≤ 10, **boundary**	the number of small angles observed at that level, over all the experiments, that originate from triangles with at least one boundary node, divided by the number of experiments.
intersection: **average of mean**	in each experiment, the average intersection number of elements at this level is computed. Average of this averages is then displayed.
intersection: **average of max**	each element at level $l+1$ intersects a certain number of level l elements. The maximum intersection observed at this coarsening level is averaged over the number of experiments.

Table 8.2: Format of Table 8.3 and latter tables

depth of shortest-distance spanning tree															
CRACK				AIRFOIL 1				AIRFOIL 2				AIRFOIL 3			
$\mathcal{C}=4$		$\mathcal{C}=8$		$\mathcal{C}=4$		$\mathcal{C}=8$		$\mathcal{C}=4$		$\mathcal{C}=8$		$\mathcal{C}=4$		$\mathcal{C}=8$	
mx	av	mx	av	mx	av	mx	av	mx	av	mx	av	mx	av	mx	av
107	52	108	53	74	20	74	22	39	8	51	14	86	18	121	42
71	34	71	35	35	15	34	14	42	13	45	16	59	22	61	22
65	31	55	28	25	11	21	10	32	13	25	11	46	18	44	18
55	28	37	19	16	7	6	3	25	11	11	6	38	14	23	11
41	22	19	10	9	4	—	—	15	8	3	1	28	13	14	7
31	16	7	3	4	2	—	—	7	4	—	—	19	9	6	2
19	10	—	—	—	—	—	—	3	1	—	—	14	6	—	—
11	5	—	—	—	—	—	—	—	—	—	—	7	3	—	—
5	2	—	—	—	—	—	—	—	—	—	—	—	—	—	—

8.4.2 Selecting the new nodes

Once the coarsening function f values are computed, a new node-set conforming to the coarsened function spacing has to be selected, such that for any two nodes i and j selected:

$$\frac{f(i)}{\beta} + \frac{f(j)}{\beta} \leq d_G(i,j)$$

To ensure this condition, the algorithm starts a search of the graph from a candidate or a selected node i, and inspects all nodes within a distance of $2f(i)/\beta$ from i. Theorem 7.3.1 proved that at most a constant number of nodes are inspected in that search. This constant depends on the coarsening constant, but the theorem showed this constant is at most $O(\mathcal{C}^4)$. It is important to know the constant in practice, since the work the algorithm performs strongly depends on it.

It should be noted that the search procedure implemented and measured here varies slightly from the one described in Section 7.3. There, once a node i is selected into the MIS, a search of of all nodes within a distance $3f(i)/\beta$ from node i is perforemd, invalidating all nodes encountered whose balls intersect with i's. In here, a search of distance $2f(i)/\beta$ is performed instead. This smaller search distance has the complication that node i is guaranteed to discover only nodes with a smaller ball, so a more careful procedure for invalidating nodes is necessary. In the search from node i, node j with a smaller ball might be encountered, such that j has been already selected for the MIS but did not invalidate i because i was not in its own search sphere. In that case, i is not added to the MIS and invalidates no nodes. Note that the intersection between i and j is guaranteed to be detected by at least one of the nodes. It our intention to implement the simpler search procedure in newer versions of the algorithm.

Ten tests each were run for each of the four meshes and two coarsening factors: $\mathcal{C} = \{4, 8\}$, for a total of eighty experiments. The spacing constant β is fixed at $\beta = 25$ during the experiments.

The table below reports the results. Each row of the table corresponds to a level in the coarsening sequence. The *max* corresponds to the maximal number of nodes inspected within a search starting from a node. The *avg* column corresponds to taking the maximum over the ten experiments of the average number of nodes within the search spheres.

number of nodes inspected in a search from a node															
CRACK				AIRFOIL 1				AIRFOIL 2				AIRFOIL 3			
$\mathcal{C}=4$		$\mathcal{C}=8$		$\mathcal{C}=4$		$\mathcal{C}=8$		$\mathcal{C}=4$		$\mathcal{C}=8$		$\mathcal{C}=4$		$\mathcal{C}=8$	
mx	av	mx	av	mx	av	mx	av	mx	av	mx	av	mx	av	mx	av
69	5	69	5	55	5	108	4	39	5	68	4	61	4	132	4
41	7	77	6	41	5	82	4	43	5	96	4	52	5	93	5
39	7	86	6	41	5	91	4	45	5	94	5	44	5	102	5
45	6	97	5	47	4	54	12	43	5	86	4	46	6	99	6
42	5	89	4	47	6	—	—	42	5	19	7	47	6	99	9
42	5	81	3	23	12	—	—	43	4	—	—	52	6	27	15
41	4	—	—	—	—	—	—	18	6	—	—	53	7	—	—
34	3	—	—	—	—	—	—	—	—	—	—	49	14	—	—

8.4.3 Protecting the edges

To make sure interior nodes do not get too close to the boundary edges, a search of the graph is performed once a boundary edge is formed. This search is very similar to the search performed from a node. Theoretically, a constant number of nodes is inspected per edge. Again, it is important to estimate this constant in practice.

Ten tests each were run for each of the four meshes and two coarsening factors: $\mathcal{C} = \{4, 8\}$, for a total of eighty experiments. The spacing constant β is fixed at $\beta = 25$ during the experiments.

The table below reports the results. Each row of the table corresponds to a level in the coarsening sequence. The *max* corresponds to the maximal number of nodes inspected within a search starting from a edge. The *avg* column corresponds to taking the maximum over the ten experiments of the average number of nodes within the search sphere of an edge. However, this average is done over the number of edges, which is much smaller than the number of nodes. Dividing the number of nodes inspected by the number of edges consistently is smaller than one (but for occasionally the last level) , and hence the total cost of this phase is negligible.

number of nodes inspected in a search from a node															
CRACK				AIRFOIL 1				AIRFOIL 2				AIRFOIL 3			
$\mathcal{C}=4$		$\mathcal{C}=8$		$\mathcal{C}=4$		$\mathcal{C}=8$		$\mathcal{C}=4$		$\mathcal{C}=8$		$\mathcal{C}=4$		$\mathcal{C}=8$	
mx	av	mx	av	mx	av	mx	av	mx	av	mx	av	mx	av	mx	av
40	14	51	15	41	8	96	11	38	11	98	23	56	19	129	30
18	5	48	7	37	7	62	22	56	15	144	27	51	9	75	14
31	6	16	4	37	7	62	22	41	13	81	14	39	8	78	15
17	4	11	5	40	10	53	18	49	11	65	23	40	9	25	5
8	4	19	9	38	16	—	—	16	7	17	10	32	6	36	12
10	5	65	44	27	10	—	—	33	14	—	—	11	4	46	10
11	6	—	—	—	—	—	—	16	10	—	—	17	6	—	—
16	10	—	—	—	—	—	—	—	—	—	—	63	13	—	—
27	18	—	—	—	—	—	—	—	—	—	—	31	8	—	—

8.5 Coarsening uniform meshes

Experiments were run on million node grids (a 1000×1000 grid). For grids of aspect ratio one, the appropriate coarsening factor is $\mathcal{C} = 2$, which leads to size reduction similar to the MIS-method. At least five experiments were performed for each method, and each of the following grid aspect ratios: $\{1, 10, 20\}$. (Recall, GRID(a) indicates a grid of triangulated rectangles, with side lengths 1 and a.)

Coarsening grid(1): Referring to Table 8.3 (Table 8.2 explains the table format), function-based coarsening is compared to MIS-based coarsening. Both methods offer very close reduction rate in the sizes of the coarsening sequence meshes, see also Figure 8.6. Similar element intersection rates are also apparent. Figure 8.6(a) plots the size of the average mesh per level in a log-scale.

The angle quality seems to be better with the function-based coarsening case, and this holds for each column of the table separately: the smallest angles, average smallest angles, boundary angles and non-boundary angles. Figure 8.6(b) plots the overall smallest angle at a level versus the average mesh size at that level. The angles of both meshes, however, are bounded away from $10°$, and both can be considered of good quality.

Coarsening grid(10) and grid(20): The behavior of MIS-based coarsening and function-based coarsening over the grids with larger aspect ratio clearly differ from the case GRID(1). Function-based coarsening produces sequences with high angle quality, at the cost of slower size reduction. The MIS-based method coarsens faster, but without sensitivity to the angle quality. See Figures 8.6(c-f), and Table 8.4. The MIS considers only the topology of the mesh, and initially, the mesh is topologically identical to that of GRID(1). The Delaunay triangulation used to obtain the coarsened mesh from the the MIS-coarsened point set restores geometrical information to some extent, but not competitively with the function-based method. It is therefore clear why the size reduction of GRID(10) and GRID(1) by the MIS-based method is similar to that of GRID(1) by the same method. For the function-based method, at least in the first few level, mesh nodes on alternating rows do not have intersecting function balls. This explains the fact the size reduction is at half rate compared to the uniform grid.

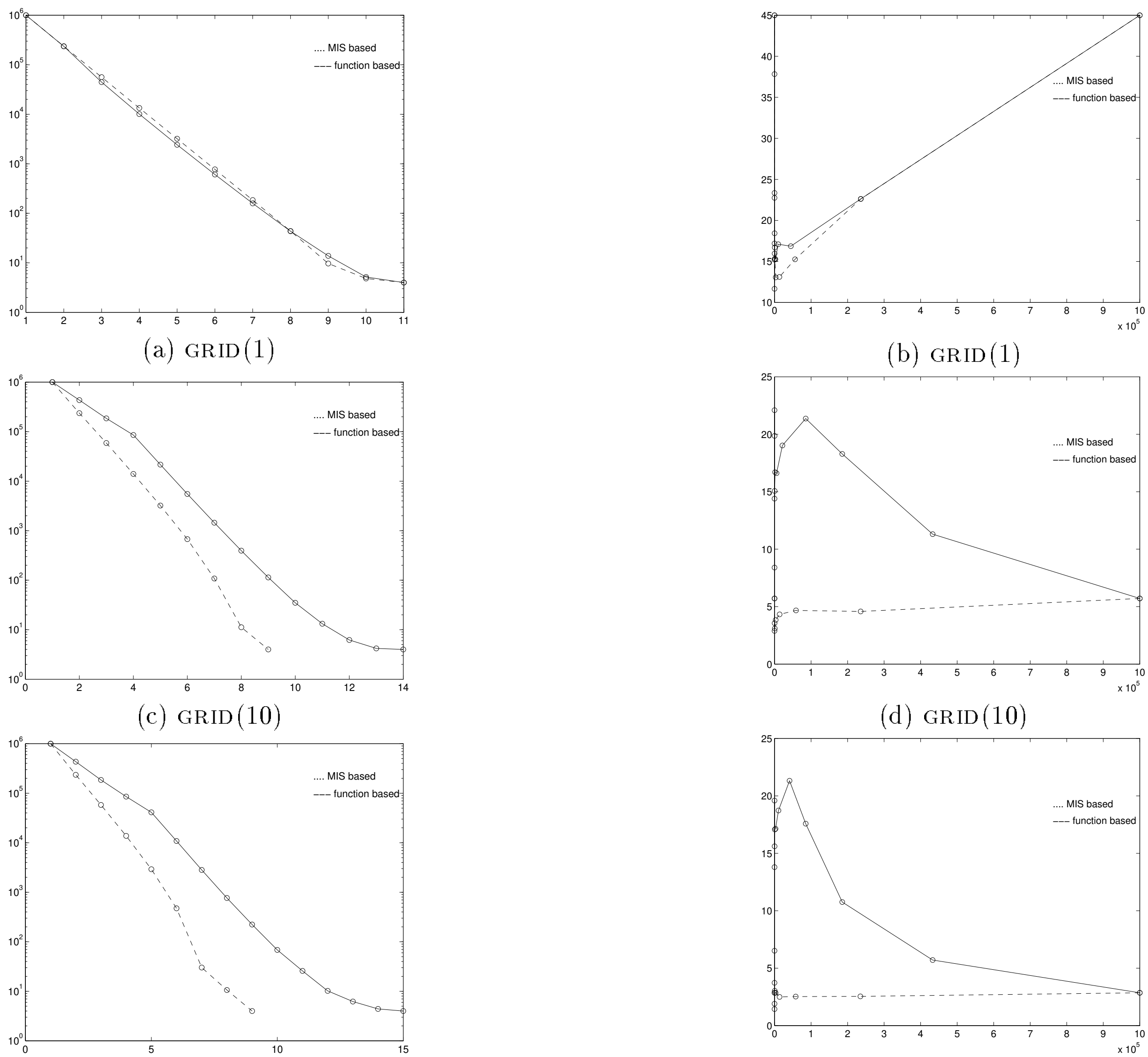
8.6 Coarsening the non-uniform meshes

Thirty experiments were run on each non-uniform mesh and each algorithm. The results are in the tables below, and Figures 8.7, 8.8 and 8.9. Looking at them, the following can be observed:

size reduction: In general, function-based coarsening reduces the size of the mesh almost as well as the MIS-based coarsening does. Figures 8.7 and 8.8 plots the size reduction of both. It is interesting to notice that the CRACK mesh is somewhat of an exception. In the first coarsening level, the function-based method reduces as much as the MIS, but in later iterations the function-based reduction progresses much slower. The theory behind function-based coarsening predicts that the amount of coarsening that can be done depends on the gradation of the mesh. Referring back to Table 8.1 it is noticeable that the CRACK mesh is the most graded one, as measured by the logarithm of its longest edge to its shortest. Especially when the size of the mesh is taken into account, it seems the mesh is extremely graded for its size. This may explain the slower size reduction.

intersection: looking at the average intersection column of each of the tables, it is noticeable that it is smaller for the function-based coarsening. This has important implications for the work

Mesh sizes **Smallest mesh angle**



(a) GRID(1) (b) GRID(1)

(c) GRID(10) (d) GRID(10)

(e) GRID(20) (f) GRID(20)

Figure 8.6: Comparing size reduction and angle quality of the function-based method versus the MIS-based method, for the three grids: GRID(1), GRID(10) and GRID(20). The left side column plots the mesh sizes versus the coarsening level, whereas the right column plots the smallest angle of the mesh versus the mesh size.

function-based: six experiments with $\beta = 25$ and $\mathcal{C} = 2$

level	number of nodes	angles						intersection average of	
		non-boundary		boundary		≤ 10			
		smallest	avg small.	smallest	avg small.	non-	boundary	mean	max
0	1000000	45.00	45.00	45.00	45.00	0	0	—	—
1	236187	22.62	22.62	22.62	22.62	0	0	8.55	22.83
2	44556	16.86	18.30	18.43	20.12	0	0	11.41	30.83
3	10088	19.35	19.82	17.10	18.72	0	0	9.92	24.17
4	2416	20.50	21.34	15.26	18.18	0	0	9.51	22.50
5	607	21.62	22.70	16.70	20.54	0	0	9.19	20.33
6	159	23.95	24.89	18.43	20.11	0	0	9.02	18.83
7	43	26.92	28.23	17.19	21.83	0	0	8.58	16.00
8	19	32.75	38.81	23.35	29.30	0	0	7.14	13.33
9	5	—	—	22.75	33.94	0	0	5.46	7.83
10	4	—	—	45.00	45.00	0	0	1.58	2.17

MIS-based: six experiments

level	number of nodes	angles						intersection average of	
		non-boundary		boundary		≤ 10			
		smallest	avg small.	smallest	avg small.	non-	boundary	mean	max
0	1000000	45.00	45.00	45.00	45.00	0	0	—	—
1	236120	22.62	22.62	22.62	22.62	0	0	8.55	23.17
2	56206	15.26	17.23	18.43	18.43	0	0	9.42	25.50
3	13421	17.03	18.20	13.11	15.96	0	0	9.54	23.33
4	3205	16.65	18.71	13.06	16.27	0	0	9.56	21.67
5	771	18.60	20.70	15.34	16.64	0	0	9.52	21.00
6	185	15.97	21.93	16.32	17.41	0	0	9.51	18.67
7	41	21.67	25.50	15.21	18.01	0	0	9.60	17.50
8	10	16.47	33.05	11.67	16.46	0	0	10.94	17.83
9	5	—	—	37.83	39.67	0	0	3.96	4.67
10	4	—	—	45.00	45.00	0	0	1.00	2.00

Table 8.3: Mesh GRID(1.0): comparing MIS-based coarsening and function-based coarsening

function-based: five experiments with $\beta = 25$ and $\mathcal{C} = 2$

level	number of nodes	angles						intersection	
		non-boundary		boundary		≤ 10		average of	
		smallest	avg small.	smallest	avg small.	non-	boundary	mean	max
0	1000000	5.71	5.71	5.71	5.71	9940090	39920	—	—
1	432944	11.31	11.31	11.31	11.31	0	0	4.35	12.00
2	185342	18.29	18.29	18.29	19.11	0	0	4.86	17.20
3	85310	21.37	21.91	25.87	26.91	0	0	4.28	15.40
4	21594	19.22	19.88	19.03	20.90	0	0	8.41	20.80
5	5518	20.19	21.76	16.62	18.73	0	0	8.74	22.00
6	1450	21.04	22.28	16.70	18.47	0	0	8.75	20.40
7	394	22.36	23.39	15.07	19.25	0	0	8.57	18.60
8	114	23.94	27.17	19.86	21.43	0	0	8.02	16.40
9	35	—	—	14.40	21.69	0	0	7.26	13.60
10	13	—	—	22.10	27.15	0	0	5.16	9.60
11	6	—	—	8.40	10.04	0	7	4.39	9.60
12	4	—	—	5.71	6.57	0	8	1.93	2.80
13	4	—	—	5.71	5.71	0	2	1.00	2

MIS-based: five experiments

level	number of nodes	angles						intersection	
		non-boundary		boundary		≤ 10		average of	
		smallest	avg small.	smallest	avg small.	non-	boundary	mean	max
0	1000000	5.71	5.71	5.71	5.71	9940090	39920	—	—
1	235736	5.10	5.21	4.58	4.69	157947	8437	12.92	75.40
2	58795	6.43	6.73	4.67	4.93	1212	1330	10.32	46.60
3	13975	5.71	6.83	4.33	4.88	22	229	9.65	54.20
4	3215	5.85	7.62	3.84	5.26	11	99	9.83	56.80
5	675	3.40	7.13	3.10	5.22	11	41	10.66	65.20
6	108	5.48	7.95	3.56	4.75	9	38	13.85	61.40
7	11	—	—	2.89	3.62	0	33	29.76	54.80
8	4	—	—	5.71	5.71	0	10	5.10	6.80

Table 8.4: Mesh GRID(10.0): comparing MIS-based coarsening and function-based coarsening

involved in the multi-grid method, since values from one level are transfered to the next using the set of elements intersected. It is also interesting to note that often the value of the maximal intersection is larger in the function-based coarsening than in the MIS-based method. As the average is much smaller, it means there are a few elements that intersect more fine elements than the rest. We have no explanation for this phenomenon.

angles of the coarsening sequence: Function-based coarsening clearly produces superior coarsening sequences with regard to angles quality. The smallest angle per level columns are much larger than the corresponding columns of the MIS-based method; similarly for the averaged smallest angle. When small angles (smaller than 10°) do appear using the function-based coarsening they are almost without exception attributable to the boundaries, as the "non/- boundary" angles columns show. Figure 8.9 plots the smallest (overall) angle per mesh size. The function-based method tends to produces sequences with smallest angle in the range 15°-20°, independently of the smallest angle of the original mesh. If the original mesh has small angles, it is improved. If the original mesh is of higher quality, such as the CRACK mesh, the angles level off at the 15°–20° range. The MIS-based method, however, performs much worse. We conclude that the function-based coarsening fulfills its promise of providing good aspect ratio coarsening sequences, subject to the boundary coarsening limitations.

crack: For ease of readability this paragraph is also attached to Table 8.5. The CRACK is a mesh of simple boundaries. This is evident in the high quality coarsening the function-based coarsening produces, see the table below. The mesh is coarsened successfully to its special points, the square end-points, while maintaining minimal angle larger than 15°. The MIS-based coarsening, on the other hand, always produces a sequence with a few bad angles. The MIS sequence is of a somewhat smaller size, see Figure 8.7, but with higher average intersection. As pointed above, the slower size reduction is probably because the mesh is very graded compared to the other meshes.

airfoil 1: For ease of readability this paragraph is also attached to Table 8.6. This is the first graded mesh with complicated boundaries covered in this section. Function-based coarsening coarsens this mesh very well. The only small angles observed is when the mesh is down to about fifty nodes, and are obviously introduced because of reaching the limits of coarsening the boundaries (the holes are too close to each other compared to the local feature size.) As the multi-grid method is anyhow likely to stop at fifty points before recursing any further, moving to a direct or iterative solver over that mesh, this is a successful coarsening. Contrast that with the number of small angles and their connection to the boundary for the MIS-based coarsening. Also, using Figure 8.7, notice the similar size reductions.

airfoil 2: For ease of readability this paragraph is also attached to Table 8.7. Mesh AIRFOIL 2 is similar to AIRFOIL 1 in its coarsening behavior. Function-based coarsening coarsens with no angles smaller than 16° (but for a few smaller angles that appear in the intial mesh as well), until the mesh is coarsened down to around twenty nodes. Small angles appear only on triangles with a boundary node. Contrast with the magnitudes and number of small angles in the MIS-based method.

As for to intersection properties: note that in the function based method, the average intersection at level four is fairly large, being about twenty five. This is well correlated with the sudden size reduction between level three and four, by a factor of about ten. Recall that coarsening is done with a coarsening factor of eight. This is helps when the mesh is graded, but once the mesh is coarsened to a fairly uniform mesh, using a high coarsening factor will cause a high reduction rate. In a more sophisticated implementation of function-based coarsening, this can be tuned. The mesh can be checked for uniformity. e.g. by looking at the edge lengths, and the coarsening factor reduced to four.

airfoil 3: For ease of readability this paragraph is also attached to Table 8.8. This mesh has the most complicated boundaries in our test suite, being also the largest non-uniform in it. By the fourth iteration, small angles appear in the function-based coarsening sequence. This is caused solely by the difficulty of coarsening the boundaries, as the relevant column shows. The wing parts are too close to each other compared to the edge-lengths at this level, so no good aspect ratio coarsening is feasible. Note however, that the MIS-based coarsening fares much worse: the first iteration already produces an angle of size $4.4°$ - while the mesh still contains three thousand points. Also, many small angles show up that are not directly attached to the boundaries. The advantages of the function-based coarsening are clearly demonstrated.

8.7 Conclusions

The ultimate test of what works and what does not is plugging the method in question into the target application and observing improved performance. The test still needed is therefore to take the coarsening-sequence we derive, and the coarsening-sequence the MIS-based coarsening produces, use them both in a multi-grid based PDE solution, and compare the performance of the resulting numerical method. This is something we have not done; we believe that the qualitative comparison of this chapter is strong evidence there are no advantages for the MIS but for its simplicity.

There are two important criteria to evaluate the performance of the resulting numerical method: the work it performs and the quality of the results it produces (the convergence behavior). The work depends on the number of levels in the coarsening sequence, the number of elements and nodes in each level, and the relative intersection of elements between levels. We believe the results of this chapter show the work involved is roughly the same; the higher quality (in terms of angles) of the function-based coarsening sequence leads us to believe that the convergence behavior will be better.

Coarsening the crack mesh

crack function-based coarsening: 30 experiments with $\beta = 25$ and $\mathcal{C} = 8$

level	number of nodes	angles						intersection	
		non-boundary		boundary		≤ 10		average of	
		smallest	avg small.	smallest	avg small.	non-	boundary	mean	max
0	10183	37.01	37.01	44.94	44.94	0	0	—	—
1	3635	15.07	18.32	18.43	23.88	0	0	5.37	46.27
2	2379	17.10	18.36	18.43	20.83	0	0	2.44	42.00
3	1300	14.03	18.01	18.43	22.38	0	0	3.06	46.60
4	466	17.13	19.92	16.70	21.71	0	0	5.39	54.57
5	79	17.35	25.21	18.43	24.64	0	0	12.22	50.70
6	6	—	—	45.00	45.00	0	0	35.88	45.93

crack MIS-based coarsening: 30 experiments

level	number of nodes	angles						intersection	
		non-boundary		boundary		≤ 10		average of	
		smallest	avg small.	smallest	avg small.	non-	boundary	mean	max
0	10183	37.01	37.01	44.94	44.94	0	0	—	—
1	3028	13.83	14.43	8.13	12.56	0	0.06	7.93	20.30
2	726	3.97	10.38	4.40	9.68	0.5	0.46	8.74	19.30
3	175	3.22	7.98	7.77	12.99	1.26	0.13	8.83	18.73
4	42	3.46	8.32	8.35	14.08	1.2	0.13	8.98	17.83
5	12	2.29	8.79	3.61	13.87	1.1	0.23	8.50	14.27
6	7	30.12	30.12	4.69	39.89	0	0.23	5.91	7.67
7	6	—	—	45.00	45.00	0	0	1.85	2.75

Table 8.5: The CRACK is a mesh of simple boundaries. This is evident in the high quality coarsening the function-based coarsening produces, see the table below. The mesh is coarsened successfully to its special points, the square end-points, while maintaining minimal angle larger than $15°$. The MIS-based coarsening, on the other hand, always produces a sequence with a few bad angles. The MIS sequence is of a somewhat smaller size, see Figure 8.7, but with higher average intersection. As pointed above, the slower size reduction is because the mesh is very graded compared to the other meshes.

Coarsening the airfoil 1 mesh

airfoil 1 function-based coarsening: 30 experiments with $\beta = 25$ and $\mathcal{C} = 8$

level	number of nodes	angles						intersection average of	
		non-boundary		boundary		≤ 10			
		smallest	avg small.	smallest	avg small.	non-	boundary	mean	max
0	4253	17.20	17.20	8.18	8.18	0	1	—	—
1	1583	17.25	19.40	17.97	20.31	0	0	4.48	78.87
2	462	20.32	22.73	16.66	21.50	0	0	6.30	52.60
3	51	23.26	31.31	6.93	9.68	0	0.33	18.27	47.77
4	13	—	—	2.62	2.62	0	4	8.04	23.03

airfoil 1 MIS-based coarsening: 30 experiments

level	number of nodes	angles						intersection average of	
		non-boundary		boundary		≤ 10			
		smallest	avg small.	smallest	avg small.	non-	boundary	mean	max
0	4253	17.20	17.20	8.18	8.18	0	1	—	—
1	1028	8.89	12.30	8.22	11.95	0.1	0.2	8.65	17.40
2	259	4.31	9.08	5.84	10.16	0.93	0.73	8.63	19.17
3	62	3.12	9.56	0.21	0.74	1.06	8.93	9.43	31.30
4	19	—	—	0.78	1.63	0	6.96	6.22	13.83
5	14	—	—	2.62	2.62	0	4.1	1.46	3.83
6	13	—	—	2.62	2.62	0	0.4	0.94	1.00

Table 8.6: This is the first graded mesh with complicated boundaries covered in this section. Function-based coarsening coarsens this mesh very well. The only small angles observed is when the mesh is down to about fifty nodes, and are obviously introduced because of reaching the limits of coarsening the boundaries (the holes are too close to each other compared to the local feature size.) As the multi-grid method is anyhow likely to stop at fifty points before recursing any further, moving to a direct or iterative solver over that mesh, this is a successful coarsening. Contrast that with the number of small angles and their connection to the boundary for the MIS-based coarsening. Also, using Figure 8.7, notice the similar size reductions.

Coarsening the airfoil 2 mesh

airfoil 2 function-based coarsening: 30 experiments with $\beta = 25$ and $\mathcal{C} = 8$

level	number of nodes	angles						intersection average of	
		non-boundary		boundary		≤ 10			
		smallest	avg small.	smallest	avg small.	non-	boundary	mean	max
0	6691	1.60	1.60	2.65	2.65	0.2	2.13	—	—
1	1727	10.44	13.91	12.26	15.69	0	0	7.30	48.47
2	568	16.72	19.50	16.11	16.11	0	0	4.86	58.53
3	196	18.88	19.63	16.11	16.11	0	0	5.06	49.73
4	18	36.18	43.28	2.31	2.99	0	0.86	24.62	46.47
5	7	—	—	1.23	1.23	0	0.1	5.99	10.53

airfoil 2 MIS-based coarsening: 30 experiments

level	number of nodes	angles						intersection average of	
		non-boundary		boundary		≤ 10			
		smallest	avg small.	smallest	avg small.	non-	boundary	mean	max
0	6691	1.60	1.60	2.65	2.65	0.2	2.13	—	—
1	1644	4.11	6.14	5.00	9.54	5.36	0.56	8.92	31.67
2	392	2.54	7.03	5.21	9.33	2.93	1.06	9.04	22.30
3	97	1.29	4.82	0.90	3.78	4.1	4.7	8.94	19.60
4	27	2.33	9.08	0.77	1.75	1.43	7.06	8.58	17.13
5	8	—	—	0.43	0.90	0	4.7	7.25	11.17
6	7	—	—	1.23	1.23	0	3	1.18	2.40

Table 8.7: Mesh AIRFOIL 2 is similar to AIRFOIL 1 in its coarsening behavior. Function-based coarsening coarsens with no angles smaller than $16°$ (but for some a few smaller angles that appear in the initial mesh as well), until the mesh is coarsened down to around twenty nodes. Small angles appear only on triangles with a boundary node. Contrast with the magnitudes and number of small angles in the MIS-based method. As for to intersection properties: note that in the function based method, the average intersection at level four is fairly large, being about twenty five. This is well correlated with the sudden size reduction between level three and four, by a factor of about ten. Recall that coarsening is done with a coarsening factor of eight. This is helps when the mesh is graded, but once the mesh is coarsened to a fairly uniform mesh, using a high coarsening factor will cause a high reduction rate. In a more sophisticated implementation of function-based coarsening, this can be tuned. The mesh can be checked for uniformity. e.g. by looking at the edge lengths, and the coarsening factor reduced to four.

Coarsening the airfoil 3 mesh

airfoil 3 function-based coarsening: 30 experiments with $\beta = 25$ and $\mathcal{C} = 8$

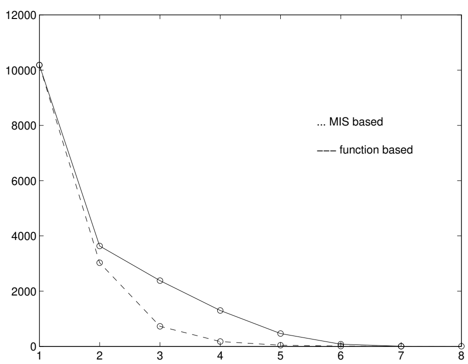
level	number of nodes	angles						intersection	
		non-boundary		boundary		≤ 10		average of	
		smallest	avg small.	smallest	avg small.	non-	boundary	mean	max
0	15606	3.58	3.58	23.67	23.67	5	0	—	—
1	2749	15.37	18.78	16.74	21.64	0	0	9.40	88.43
2	1219	18.45	21.17	16.57	20.89	0	0	3.67	56.07
3	522	20.36	23.60	2.67	4.28	0	2.1	3.74	49.53
4	186	20.98	25.08	0.66	0.66	0	10.43	5.10	48.27
5	26	39.83	48.02	0.26	0.35	0	14.1	16.71	49.03
6	15	—	—	0.08	0.08	0	15	4.31	15.93

airfoil 3 MIS-based coarsening: 30 experiments

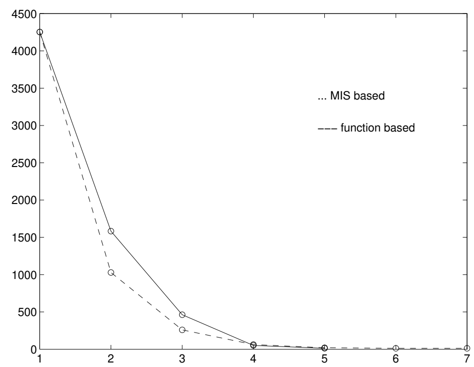
level	number of nodes	angles						intersection	
		non-boundary		boundary		≤ 10		average of	
		smallest	avg small.	smallest	avg small.	non-	boundary	mean	max
0	15606	3.58	3.58	23.67	23.67	5	0	—	—
1	3708	4.43	8.62	13.47	16.66	1.53	0	8.69	18.53
2	908	2.19	6.85	5.96	12.23	3.56	0.33	8.77	20.80
3	175	1.18	4.62	0.03	0.05	6.13	25	11.88	91.13
4	44	0.36	6.14	0.08	0.08	2.1	15.86	8.63	23.27
5	17	—	—	0.08	0.08	0	14.86	4.87	14.23
6	15	—	—	0.08	0.08	0	15	1.33	4.50

Table 8.8: This mesh has the most complicated boundaries in our test suite, being also the largest non-uniform in it. By the fourth iteration, small angles appear in the function-based coarsening sequence. This is caused solely by the difficulty of coarsening the boundaries, as the relevant column shows. The wing parts are too close to each other compared to the edge-lengths at this level, so no good aspect ratio coarsening is feasible. Note however, that the MIS-based coarsening fares much worse: the first iteration already produces an angle of size $4.4°$ - while the mesh still contains three thousand points. Also, many small angles show up that are not directly attached to the boundaries. The advantages of the function-based coarsening are clearly demonstrated.

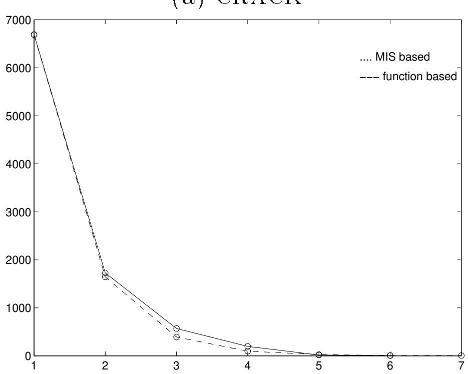
Coarsening sequence mesh size reduction



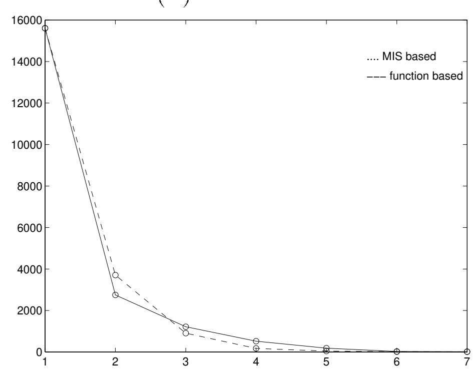
(a) CRACK



(b) AIRFOIL 1



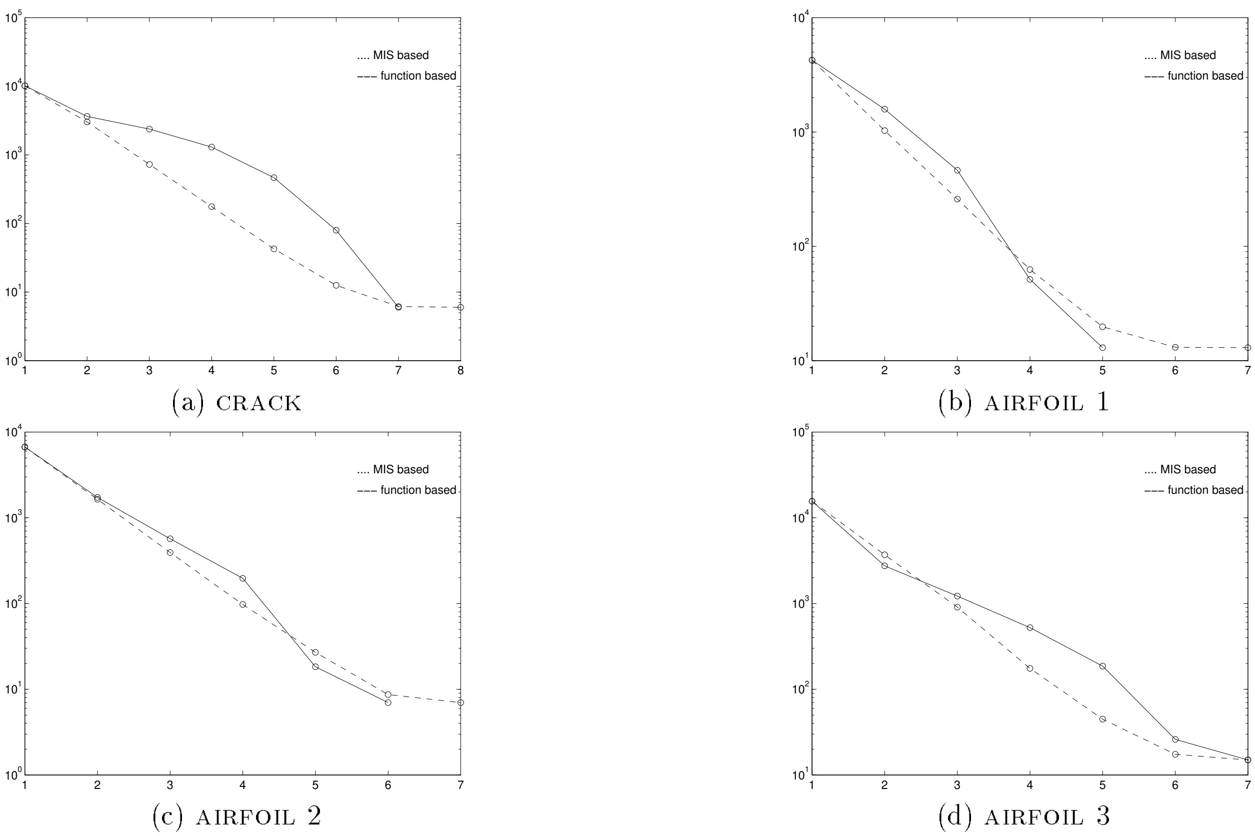
(c) AIRFOIL 2



(d) AIRFOIL 3

Figure 8.7: For each of the four meshes, the number of nodes of each mesh is plotted for each level of the coarsening sequence, both for the MIS-based coarsening and the function-based coarsening.

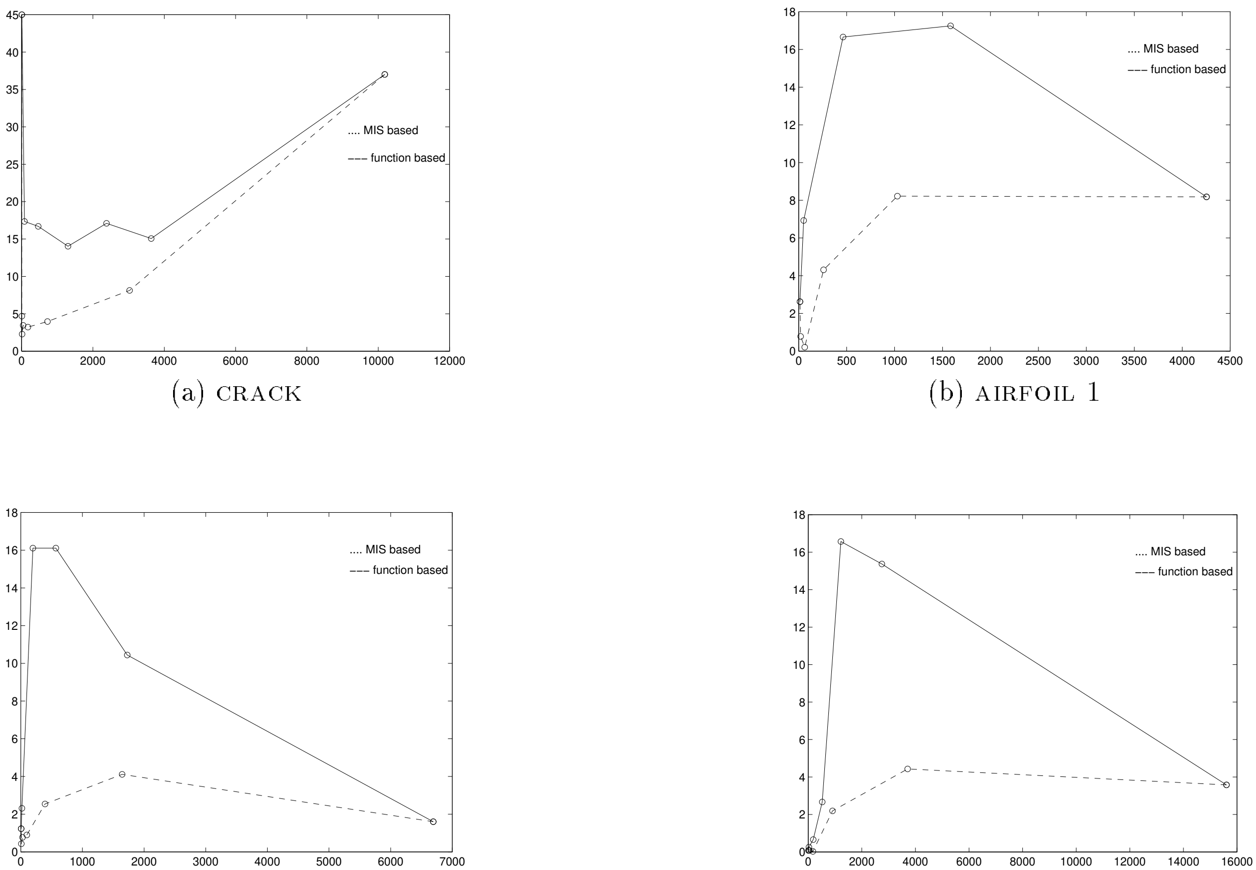
Coarsening sequence mesh size reduction - log scale



(a) CRACK (b) AIRFOIL 1 (c) AIRFOIL 2 (d) AIRFOIL 3

Figure 8.8: For each of the four meshes, the log of the number of nodes of each mesh is plotted for each level of the coarsening sequence, both for the MIS-based coarsening and the function-based coarsening.

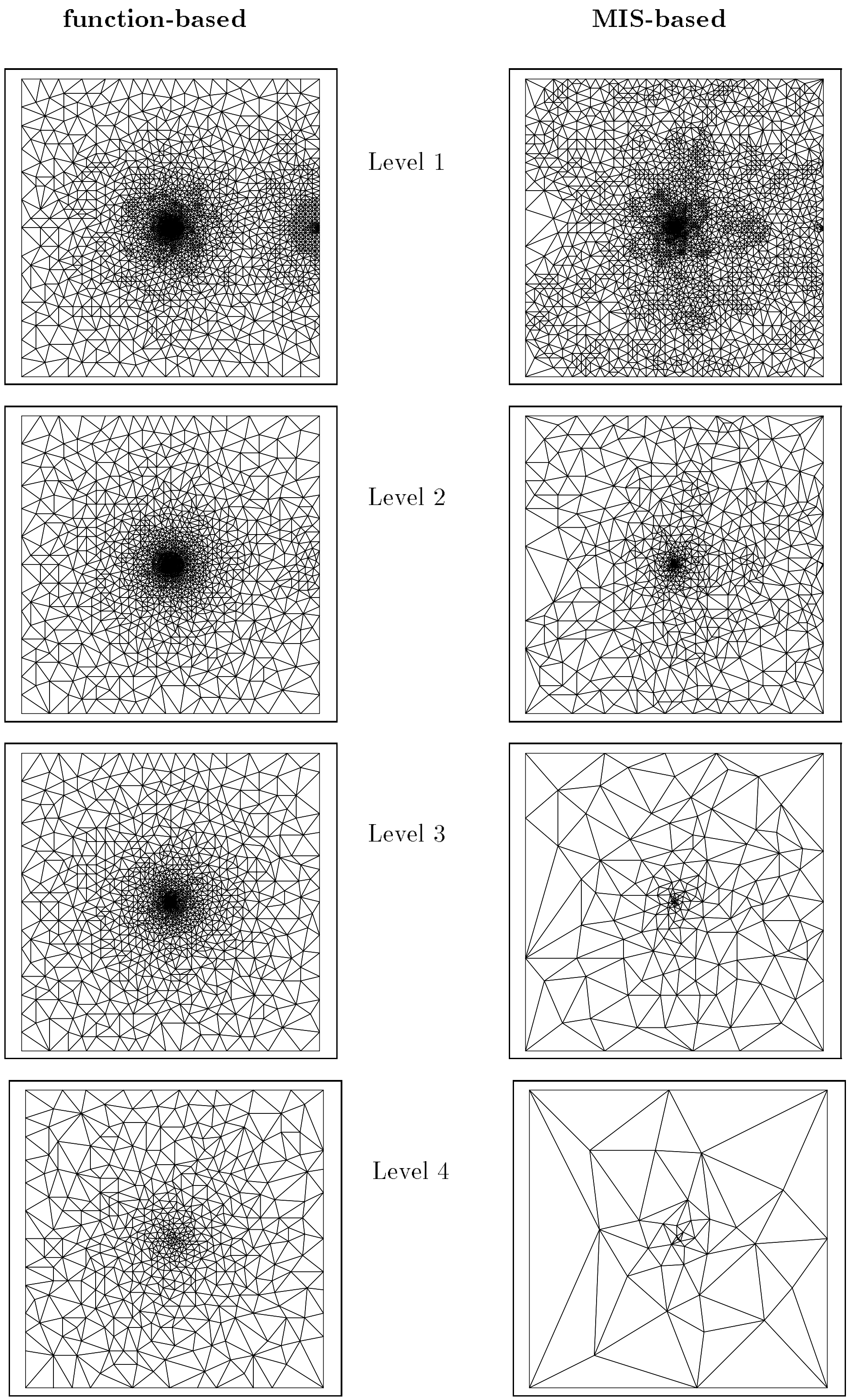
Angle quality of the coarsening sequence



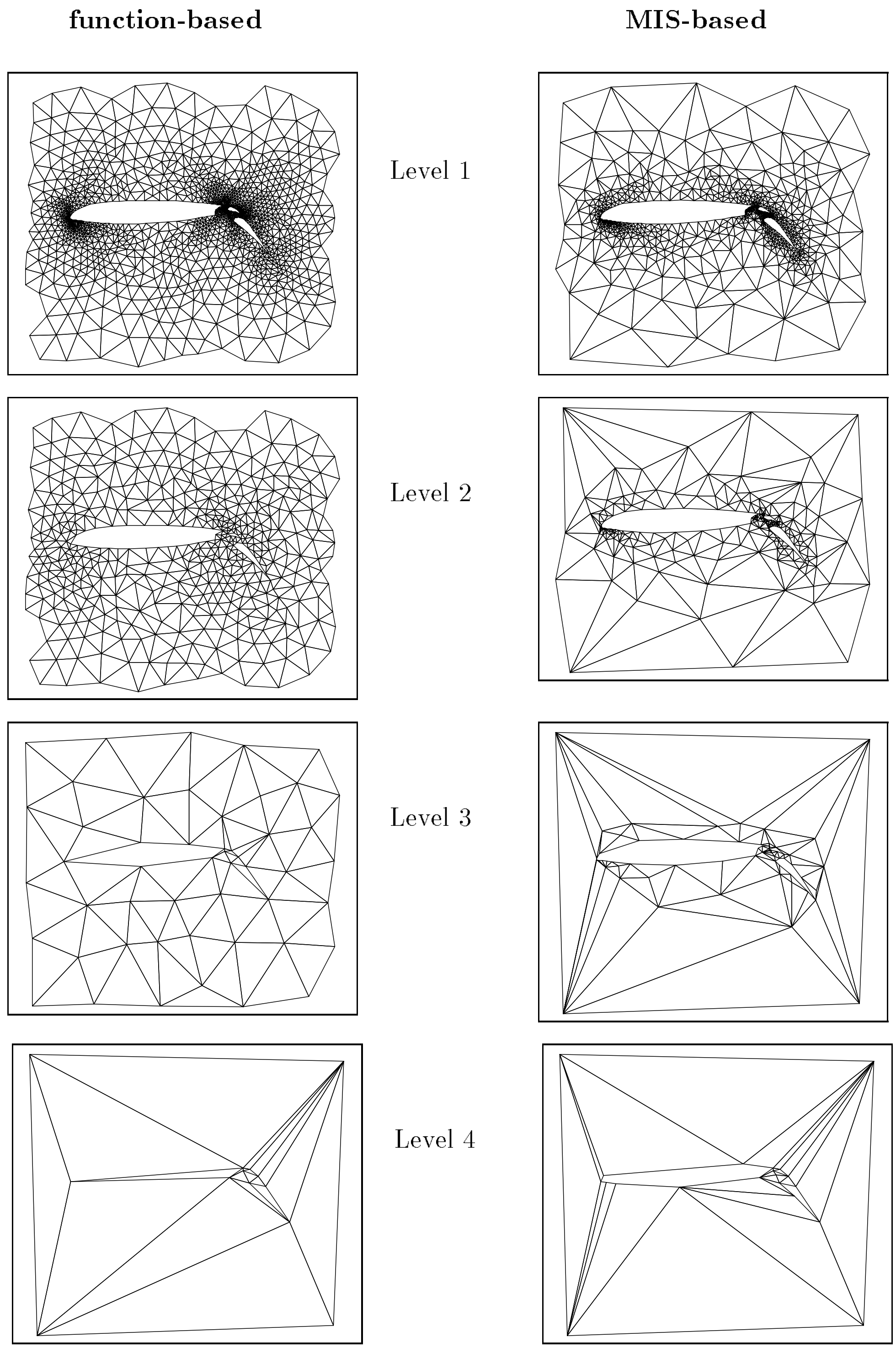
(a) CRACK (b) AIRFOIL 1 (c) AIRFOIL 2 (d) AIRFOIL 3

Figure 8.9: For each of the four meshes, the smallest angle observed at a level, plotted against the average size of the mesh at that level.

coarsening sequences for the crack mesh

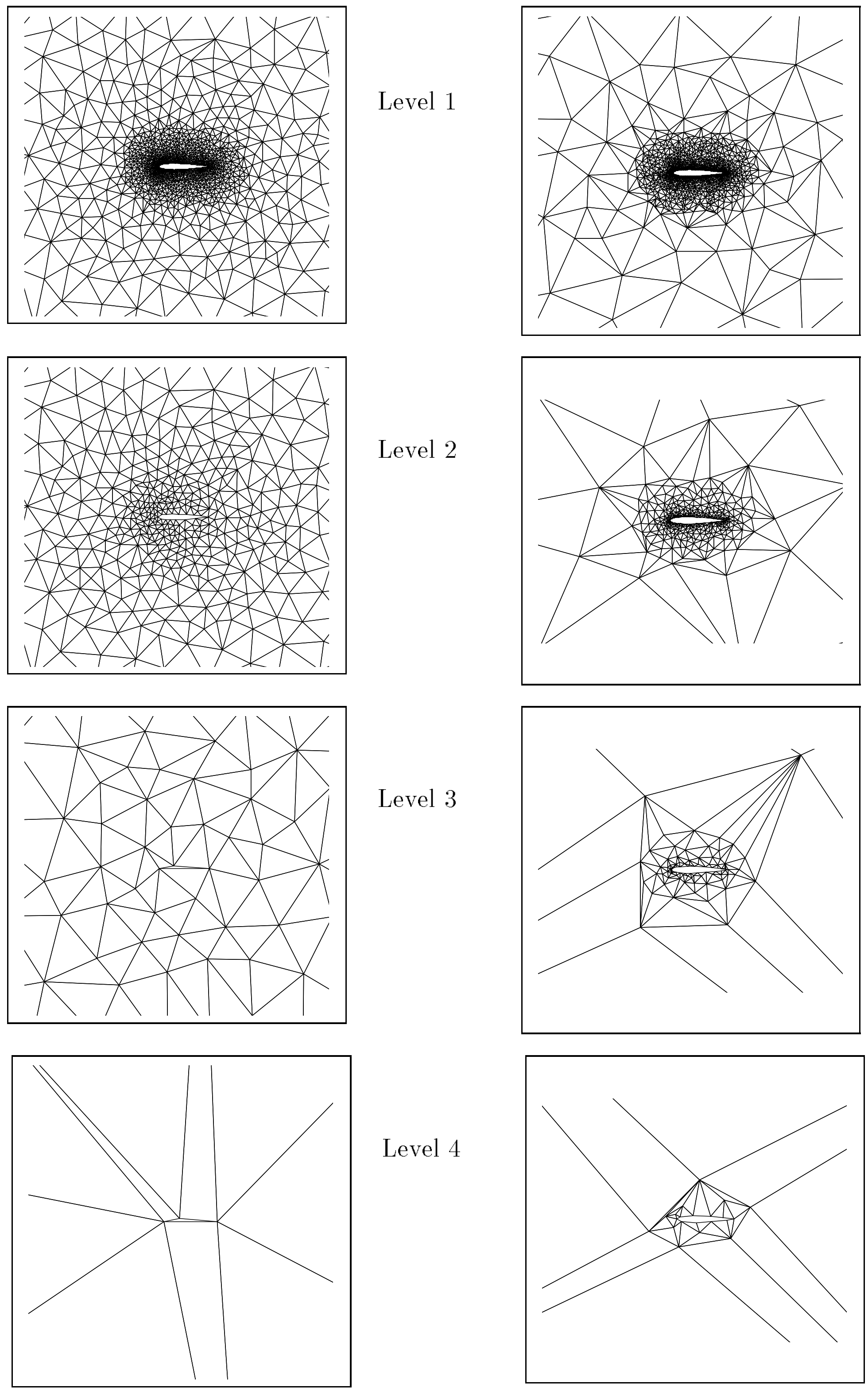


coarsening sequences for the airfoil 1 mesh



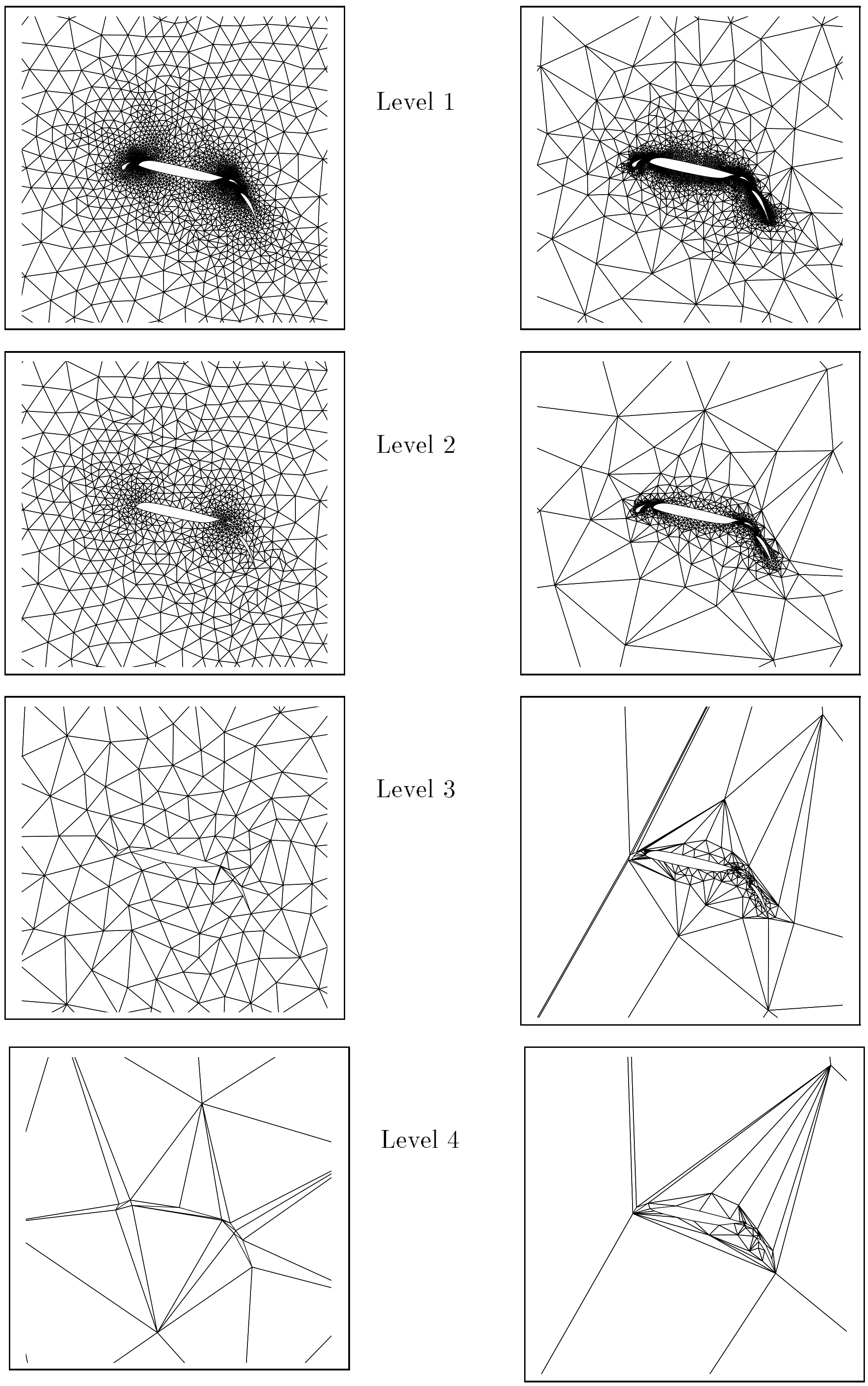
coarsening sequences for the airfoil 2 mesh

function-based **MIS-based**



coarsening sequences for the airfoil 3 mesh

function-based **MIS-based**



Chapter 9

Discussion and Future Work

In this thesis, we introduced the concept of well-spaced points, and demonstrated the importance of well-spaced points for the finite element method, for the control volume method, and for the multigrid method. The techniques used in the manipulation of well-spaced points are crucial to Delaunay-based mesh generation and coarsening.

Section 1.3 outlines the results of this thesis in some detail. In this section, we point out some of the questions that were left unanswered along the way, and discuss a future extension of our mesh coarsening algorithm.

9.1 Well-spaced points and bounded aspect ratio meshes

The relationship between well-spaced points and bounded aspect ratio meshes was discussed in detail in Chapters 2, 3 and 4. In two dimensions, a well-spaced point set is equivalent to a bounded aspect ratio mesh. If we neglect the boundaries in this short discussion, this equivalence means that the nodes of a bounded aspect ratio mesh in two dimensions are well-spaced, and that conversely, given a well-spaced point set we can generate a bounded aspect ratio mesh from this set, by, for example, applying the Delaunay triangulation. Note the important role the Delaunay triangulation plays in showing this equivalence.

In three dimensions, the nodes of a bounded aspect ratio mesh are well-spaced as well. However, we did not show the converse direction. The problem is that the Delaunay triangulation of a well-spaced point set in three dimensions fails to produce a bounded aspect ratio mesh. The culprit is a tetrahedron called a sliver; see Section 3.3. A sliver is a very flat tetrahedron whose circumradius is of the same scale as its edges. Such a tetrahedron can easily arise in the Delaunay triangulation of well-spaced points. That does not necessarily imply that well-spaced points have no bounded aspect ratio tetrahedralization in three-dimensions. Rather, it implies that the Delaunay triangulation is the wrong tool for the task. An interesting open question is: does every well-spaced point set in three dimensions lead to a bounded aspect ratio mesh, or is there an example of a well-spaced point set such that none of its tetrahedralizations is of bounded aspect ratio? we may consider infinite domains (perhaps with a periodic point set), or, to bring boundaries back into the picture, a simple domain in the shape of a unit cube. In the latter case, the point set must also be well-spaced with respect to the lower dimensional elements of the cube: its faces and edges. Furthermore, restrictions must be placed on the relative location of interior points and the cube faces; if an interior point is too close to a face, relative to the spacing function, a bad radius-edge ratio tetrahedron may form.

9.2 Size optimal bounded radius-edge meshes

Well-spaced points in three dimensions might give rise to a Delaunay triangulation that contains slivers. In order to bypass this problem, we defined a new aspect ratio, the radius-edge aspect ratio, and a new class of meshes, the bounded radius-edge ratio Delaunay meshes. Chapter 4 gives an algorithm to generate a bounded-aspect ratio Delaunay mesh conforming to a given piecewise linear non-manifold domain. However, this algorithm generates meshes whose size is comparable to the size of the optimal bounded *aspect ratio* mesh tetrahedralizing the same domain.

The problem is that a bounded radius-edge ratio could potentially use many fewer elements than a bounded aspect ratio mesh to decompose the same domain. Section 4.4 gave several examples for simple domain descriptions that can be tetrahedralized with a constant number of bounded radius-edge ratio tetrahedra and yet can require an arbitrary number of bounded aspect ratio tetrahedra.

The open question that remains is that of finding an algorithm that generates a provably good bounded radius-edge ratio mesh, i.e., a mesh that contains at most a constant factor more elements than the optimal bounded radius-edge ratio mesh decomposing the same domain. This problem has been discussed at some length in Section 4.4. Furthermore, the same section discusses the difficulty in defining a spacing function that is similar to the local feature size, but is intended for bounded radius-edge ratio meshes and captures their optimal spacing.

9.3 Multigrid method

The main application presented here of the well-spaced points technology is to the problem of mesh coarsening. Mesh coarsening is an important subroutine of the multigrid method, as was discussed in Section 1.2 and Chapter 5.

To coarsen a mesh via its well-spaced points, we use a spacing function of the points. Hence, our approach to mesh coarsening is called function-based. Chapter 5 presented our function-based approach and gave theoretical guarantees for the geometrical quality of the coarsening sequences we produce. In Chapter 8 we provide experimental evidence that our approach produces high-quality coarsening sequences in practice, as well as in theory. In particular, we show that the quality of the coarsening sequences we produce is far better than that of the popular MIS-based approach. However, the ultimate test is by plugging the method in question into the target application and observing improved performance. Therefore, the test still needed is to take the coarsening sequences we generate, and the coarsening sequences MIS-based coarsening generates, use them both in a multigrid based PDE solution, and compare the performance of the resulting numerical method.

Whereas we have not performed this test, we believe that the qualitative comparison of this thesis gives strong evidence that there are no advantages for the MIS but for its simplicity. There are two important criteria for evaluating the performance of the resulting numerical method: the amount of work it performs and the quality of the results it produces (the convergence behavior). The amount of work depends on the number of levels in the coarsening sequence, the number of elements and nodes in each level, and the relative intersection of elements between levels. We believe the results of this thesis show the work involved is roughly the same; the higher quality (in terms of angles) of the function-based coarsening sequence leads us to believe that the convergence behavior will be better as well.

9.4 Adaptive meshing

An exciting future application for this mesh coarsening work is to the problem of adaptive meshing. Adaptive meshing employs a mix of mesh coarsening and mesh refinement. An example application is when simulating an advancing front of a wave. The front of the wave should contain small elements, whereas in the wake of the wave large elements are preferable. The challenge is to quickly modify the mesh as the wave moves. In some areas the mesh then needs to be refined, and in areas where the physics no longer demand high resolution to achieve a desired degree of accuracy, it should be coarsened.

The function-based coarsening approach can be easily adapted to a mixed coarsening and refinement framework. The main change from the pure coarsening case is that the new spacing function may be smaller than the former spacing function in areas where the mesh should be refined. In these areas, new points should be added. In the next phase, which is similar to pure coarsening, points are eliminated to guarantee that the spacing function balls around the nodes are disjoint. We hope to study this approach to adaptive meshing in the future.

Bibliography

[1] I. Babuška and A.K. Aziz. On the angle condition in the finite element method. *SIAM Journal on Numerical Analysis*, 13(2):214–226, 1976.

[2] Chanderjit L. Bajaj and Tamal K. Dey. Convex decompositions of polyhedra and robustness. *SIAM Journal on Computing*, 21(2):339–364, April 1992.

[3] R. E. Bank. *PLTMG: A software package for solving elliptic partial differential equations – users' guide 6.0*, volume 7 of *Frontiers in Applied Mathematics*. SIAM Books, Philadelphia, 1990.

[4] R. E. Bank, A. H. Sherman, and A. Weiser. Refinement algorithms and data structures for regular local mesh refinement. In R. S. Stepleman, editor, *Scientific Computing: Applications of Mathematics and Computing to the Physical Sciences*, pages 3–17. North–Holland, Amsterdam, 1983.

[5] Randolph E. Bank and Jinchao Xu. An algorithm for coarsening unstructured meshes. *Numerische Mathematik*, 73:1–36, 1996.

[6] Timothy J. Barth. Randomized multigrid. *AIAA paper*, 95-0207, 1995.

[7] Eric B. Becker, Graham F. Carey, and J. Tinsley Oden. *Finite elements: an introduction*. The Texas Finite Element Series. Prentice-Hall, 1981.

[8] Marshall Bern and David Eppstein. Mesh generation and optimal triangulation. In D.-Z. Du and F. K. Hwang, editors, *Computing in Euclidean Geometry*, volume 1 of *Lecture Notes Series on Computing*, pages 23–90. World Scientific, Singapore, 1992.

[9] Marshall Bern, David Eppstein, and John Gilbert. Provably good mesh generation. In *Proceedings of the 31st Annual IEEE Symposium on the Foundations of Computer Science*, pages 231–241, 1990.

[10] Marshall Bern and Paul Plassmann. Mesh generation. In Jörg Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*. Elsevier Science Publishers, to appear.

[11] J. H. Bramble, J. E. Pasciak, and J. Xu. Convergence estimates for product iterative methods with application s to domain decomposition. *Mathematics of Computation*, 55:1–22, 1991.

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

[13] W. L. Briggs. *A Multigrid tutorial*. SIAM Books, Philadelphia, 1987.

[14] Graham F. Carey. *Computational Grids: Generation, Adaptation, and Solution Strategies.* Taylor and Francis, June 1997.

[15] T. F. Chan and B. Smith. Domain decomposition and multigrid algorithms for elliptic problems on unstructured meshes. *Contemporary Mathematics*, pages 1–14, 1993.

[16] Timothy M. Chan, Jack Snoeyink, and Chee-Keng Yap. Output-sensitive construction of polytopes in four dimensions and clipped Voronoi diagrams in three. In *Proceedings of the 6th ACM-SIAM Symposium on Discrete Algorithms*, pages 282–291, 1995.

[17] Tony F. Chan and Jun Zou. Additive Schwarz domain decomposition methods for elliptic problems on unstructured meshes. CAM Report 95-16, Department of Math, UCLA, March 1995.

[18] B. Chazelle. Convex partitions of polyhedra: a lower bound and worst-case optimal algorithm. *SIAM Journal on Computing*, 13:488–507, 1984.

[19] B. Chazelle and L. Palios. Triangulating a non-convex polytope. In *5th ACM Symposium on Computational Geometry*, pages 393–400, 1989.

[20] L. P. Chew. There is a planar graph almost as good as the complete graph. In *2nd ACM Symposium on Computational Geometry*, pages 169–177, 1986.

[21] L. P. Chew. Guaranteed-quality triangular meshes. Technical Report TR 89-983, Department of Computer Science, Cornell, Ithaca, NY, 1989.

[22] L. P. Chew. Guaranteed quality Delaunay meshing in 3D. In *13th ACM Symposium on Computational Geometry*, 1997.

[23] NASgraph collection. http://science.nas.nasa.gov/Software/GAMFF/.

[24] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* The MIT Press, Cambridge, Mass., 1990.

[25] D. P. Dobkin, S. J. Friedman, and K. J. Supowit. Delaunay graphs are almost as good as complete graphs. *Discrete and Computational Geometry*, 5:399–407, 1990.

[26] S. E. Dorward, L. R. Matheson, and R. E. Tarjan. Unstructured multigrid strategies on massively parallel computers: a case for integrated design. In *Proceedings of the 27th Annual Hawaii International Conference on System Sciences*, pages 169–178, 1994.

[27] Michael Drmota and Robert F. Tichy. *Sequences, Discrepancies and Applications*, volume 1651 of *Lecture Notes in Mathematics.* Springer–Verlag, 1997.

[28] David Eppstein. Spanning trees and spanners. Technical Report TR 96-16, University of California at Irvine, 1996.

[29] C.A.G. Fletcher. *Computational techniques for fluid dynamics.* Springer Series in Computational Physics. Springer-Verlag, 2nd edition, 1991.

[30] S. Fortune. A sweepline algorithm for Voronoi diagrams. *Algorithmica*, 2:153–174, 1987.

[31] Omar Ghattas and Xiaogang Li. Private communications, 1996.

[32] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The John Hopkins University Press, 3rd edition, 1996.

[33] L. J. Guibas, D. E. Knuth, and M. Sharir. Randomized incremental construction of Delaunay and Voronoi diagrams. *Algorithmica*, 7:381–413, 1992.

[34] H. Guillard. Node nested multigrid with Delaunay coarsening. Tech. report, INRIA, Valbonne, France, 1993.

[35] C. Gumbert, R. Lohner, P. Parikh, and S. Pirzadeh. A package for unstructured grid generation and finite element flow solvers. *AIAA paper*, 89–2175, 1989.

[36] Paul Heckbert and Michael Garland. A survey of surface simplification algorithms. Technical Report To appear, Carnegie Mellon University, 1997.

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

[38] Claes Johnson. *Numerical solution of partial differential equations by the finite element method*. Cambridge University Press, 1992.

[39] J. M. Keil and C. A. Gutwin. The Delaunay triangulation closely approximates the complete euclidean graph. In *1st workshop on algorithms and data structures*, pages 47–56, 1989.

[40] Lallemand, Steve, and Dervieux. Unstructured multigriding by volume agglomeration: current status. Technical Report 1224, Inria-Sophia Antipolis, 1990.

[41] Anwei Liu and Barry Joe. Quality local refinement of tetrahedral meshes based on bisection. *SIAM Journal on Scientific Computing*, 16:1269–1291, 1995.

[42] Y. Y. Lu, T. Belytschko, and L. Gu. A new implementation of the element free Galerkin method. *Computational Methods in Applied Mechanical Engineering*, 113:397–414, 1994.

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

[44] J. M. Maubach. local bisection refinement for n–simplicial grids generated by reflection. *SIAM Journal on Scientific Computing*, 16:210–277, 1995.

[45] D. J. Mavriplis. An advancing front Delaunay triangulation algorithm designed for robustness. *AIAA Paper*, 93–0671, 1993.

[46] D. J. Mavriplis. Multigrid techniques for unstructured meshes. Technical Report 95–27, ICASE, 1995.

[47] D. J. Mavriplis. Unstructured mesh generation and adaptivity. Technical Report 95–26, ICASE, Hampton, VA, April 1995.

[48] D. J. Mavriplis and D. Venkatakrishnan. Agglomeration multigrid for viscous turbulent flows. *AIAA paper*, 1994.

[49] Gary L. Miller, Dafna Talmor, Shang-Hua Teng, and Noel Walkington. A Delaunay based numerical method for three dimensions: generation, formulation and partition. In *Proceedings of the 27th Annual ACM Symposium on Theory of Computing*, 1995.

[50] Gary L. Miller, Dafna Talmor, Shang-Hua Teng, Noel Walkington, and Han Wang. Control volume meshes using sphere packing: Generation, refinement and coarsening. In *5th International Meshing Roundtable*, pages 47–61, 1996.

[51] Don P. Mitchell. Generating anti-aliased images at low sampling densities. *ACM Computer graphics*, 21(4):65–71, 1987.

[52] S. A. Mitchell and S. A. Vavasis. Quality mesh generation in three dimensions. In *8th ACM Symposium on Computational Geometry*, pages 212–221, 1992.

[53] W. F. Mitchell. Optimal multilevel iterative methods for adaptive grids. *SIAM Journal on Scientific and Statistical Computing*, 13:146–167, 1992.

[54] W. F. Mitchell. MGGHAT: elliptic PDE software with adaptive refinement, multigrid and high order finite elements. In N. D. Melson, T. A. Manteuffel, and S. F. McCormick, editors, *Sixth Copper Mountain Conference on Multigrid Methods*, volume CP 3224, pages 439–448, Hampton, VA, 1993. NASA.

[55] R. A. Nicolaides. Direct discretization of planar div–curl problems. *SIAM Journal on Numerical Analysis*, 29(1):32–56, 1992.

[56] Harald Niederreiter. *Random number generation and quasi-Monte Carlo methods*, volume 63 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM Books, Philadelphia, 1992.

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

[58] J. Peraire, M. Vahdati, K. Morgan, and O. C. Zienkiewicz. Adaptive remeshing for compressible flow computations. *Journal of computational physics*, 72:449–466, 1987.

[59] S. Pirzadeh. Structured background grids for generation of unstructured grids by advancing front methods. *AIAA Journal*, 31(2):257–265, 1993.

[60] V. T. Rajan. Optimality of the Delaunay triangulation in R^d. In *7th ACM Symposium on Computational Geometry*, pages 357–363, 1991.

[61] B. D. Ripley. Modelling spatial patterns. *Journal of the royal statistical society, Series B*, 39(2):172–192, 1977.

[62] M. C. Rivara. Algorithms for refining triangular grids suitable for adaptive and multigrid techniques. *Journal on Numerical Methods in Engineering*, 20:745–756, 1984.

[63] I. G. Rosenberg and F. Stenger. A lower bound on the angles of triangles constructed by bisecting the longest side. *Mathematics of Computation*, 29:390–395, 1975.

[64] J. Ruppert. A new and simple algorithm for quality 2-dimensional mesh generation. In *Proceedings of the 4th ACM-SIAM Symposium on Discrete Algorithms*, pages 83–92, 1993.

[65] J. Ruppert and R. Seidel. On the difficulty of triangulating three-dimensional non-convex polyhedra. *Discrete and Computational Geometry*, 7:227–253, 1992.

[66] E. Schönhardt. Über die Zerlegung von Dreieckspolyedern in Tetraeder. *Math. Ann.*, 98:309–312, 1928.

[67] Jonathan Shewchuk. *Delaunay refinement mesh generation.* PhD thesis, Carnegie Mellon University, 1997.

[68] Peter Shirley. Discrepancy as a quality measure for sample distributions. In *Eurographics*, pages 183–194, 1991.

[69] G. Strang. *An introduction to applied mathematics.* Wellesley-Cambridge Press, 1986.

[70] M. Stynes. On faster convergence of the bisection method for all triangles. *Mathematics of Computation*, 35:1195–1201, 1980.

[71] Shang-Hua Teng. A geometric approach to parallel hierarchical and adaptive computing on unstructured meshes. In *Fifth SIAM Conference on Applied Linear Algebra*, pages 51–57, June 1994.

[72] D. Venkatakrishnan and D. J. Mavriplis. Agglomeration multigrid for the 3D Euler equation. *AIAA paper*, 1994.

[73] D. F. Watson. Computing the n-dimensional Delaunay tessellation with applications to Voronoi polytopes. *Comput. J.*, 24(2):167–172, 1981.

[74] John I. Yellott Jr. Spectral consequences of photo-receptor sampling in the rhesus retina. *Science*, 221:382–385, 1983.