Arc Triangulations[*]

Oswin Aichholzer[†] Wolfgang Aigner[†§] Franz Aurenhammer[‡] Kateřina Čech Dobiášová[§]
Bert Jüttler[§]

Abstract

The quality of a triangulation is, in many practical applications, influenced by the angles of its triangles. In the straight line case, angle optimization is not possible beyond the Delaunay triangulation. We propose and study the concept of circular arc triangulations, a simple and effective alternative that offers flexibility for additionally enlarging small angles. We show that angle optimization and related questions lead to linear programming problems, and we define unique flips in arc triangles. A possible application of arc triangulations in the area of graph drawing is detailed.

1 Introduction

Geometric graphs and especially triangulations are an ubiquitous tool in geometric data processing [2, 8, 12]. The quality of a given triangular mesh naturally depends on the size and shape, in particular the angles, of its composing triangles. In practice, quite often the Delaunay triangulation (see, e.g., [8]) is the mesh of choice, because it maximizes the smallest angle over all possible triangulations of a given finite set of points in the plane. Still, the occurrence of 'poor' triangles cannot be avoided sometimes, especially near the boundary of the input domain, or due to the presence of mesh vertices of high edge degree.

The situation becomes different (and interesting again) if the requirement that triangulation edges be straight is dropped. In applications as finite element methods or graph drawing, the numerical and optical benefits of a graph that potentially grants nice angles can be exploited fully only if curved edges are admitted. In this paper, we try to encourage the use of so-called *arc triangulations*, which simply are triangulations whose edges are circular arcs. Modeling triangulations this way bears several advantages if angles are to be optimized. Small angles at the boundary can be enlarged by optimizing the arc curvatures for the given triangulation. Situations with vertices of high degree can be faced by applying angle-improving flips in arc triangles that reduce the vertex degree.

Maximizing the smallest angle in a combinatorially fixed arc triangulation of a point set can be formulated as a linear program (Section 2). This guarantees a fast solution of this optimization problems for arc triangulations in practice. Moreover, the linear program will tell us whether a given domain admits an arc triangulation of a pre-specified combinatorial type, by checking whether its feasible region is void. In particular, flips for arcs can be defined (Section 3), via optimization after the flip has been applied combinatorially. If we want to optimize equiangularity in an arc triangulation (i.e., maximize the sorted angle vector lexicographically) then we can do so as well. After having maximized the first k smallest angles, we keep them fixed in the linear program for maximizing the $(k+1)$st angle, for $k \geq 0$.

We believe that arc triangulations constitute a useful tool in several important application areas as finite element methods or especially graph drawing (Section 5). In view of graph drawing applications [5, 6], it is desirable to extend our approach to optimizing angles in general plane graphs. As our simple optimization method works only for full triangulations, we simply complete the graphs to suitable (e.g. Delaunay triangulation [10, 4]) triangulations and treat the newly obtained angles differently as explaind in Section 5. In several applications, the boundary of the underlying domain will be given as a polynomial spline curve. Such domains can be approximated in a convenient way using circular biarc splines [1], and thus are naturally suited to triangulation by circular arcs.

2 Angle Optimization

Consider a straight line triangulation, $\mathcal{T}$, in a given domain D of the plane. No restrictions on D are required but, for the ease of presentation, let D be simply connected and have piecewise circular (or linear) boundary. In general, $\mathcal{T}$ will use vertices in the interior of D. We are interested in the following optimization problem: Replace each interior (i.e., non-boundary) edge of $\mathcal{T}$ by some circular arc, in a way such that the smallest angle in the resulting arc triangulation is maximized. To see that this problem is well defined, notice that the optimal solution, call it $\mathcal{T}^*$, cannot contain negative angles: The smallest angle between arcs has to be at least as large as the smallest angle that arises in $\mathcal{T}$. As a consequence, for each vertex in S, the order of its incident arcs in $\mathcal{T}^*$ coincides with the order of its incident edges in the input triangulation $\mathcal{T}$. In other words, each arc triangle

[*]Supported by FWF NRN 'Industrial Geometry' S92

[†]Institute for Software Technology, Graz University of Technology, Austria, {oaich,waigner}@ist.tugraz.at

[‡]Institute for Theoretical Computer Science, Graz University of Technology, Austria, auren@igi.tugraz.at

[§]Institute of Applied Geometry, Johannes Kepler University Linz, Austria, {Bert.Juettler, Katerina.Cech-Dobiasova}@jku.at

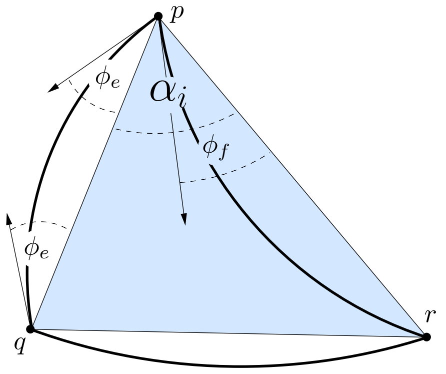


Figure 1: Angles of deviation

in $\mathcal{T}^*$ is *well-oriented*, i.e., it has the same orientation as its straight line equivalent. Therefore, no overlap of arcs or arc triangles in $\mathcal{T}^*$ can occur. Interestingly, this is a specialty of triangulations; the last conclusion remains no longer true if faces with more than three arcs are present. We postulate for the rest of this paper that arc triangles be well-oriented.

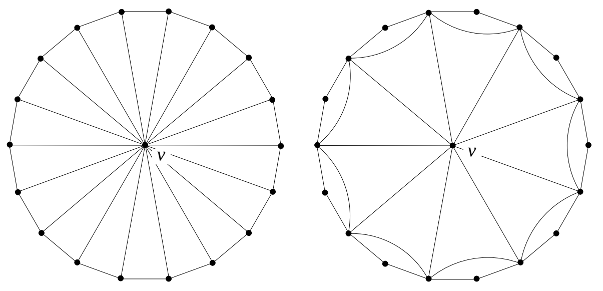
We now formulate the angle optimization problem as a linear program. For each interior edge $e = \overline{pq}$ in the triangulation $\mathcal{T}$ we introduce one variable, ϕ_e, describing the angle at which the circular arc $\widehat{pq}$ deviates from the straight connection of p and q (at these very points). Figure 1 offers an illustration. Note that ϕ_e may take on positive or negative values, depending on the sidedness of $\widehat{pq}$ with respect to e. For each edge e' of $\mathcal{T}$ on the input boundary ∂D, we fix $\phi_{e'}$ to the value $d_{e'}$ given by ∂D.[1] The inequalities for the linear program now stem from the angles α_i arising in $\mathcal{T}$. If e and f are the two edges of $\mathcal{T}$ that define α_i, we consider the angle between the two respective circular arcs, $\beta_i = \phi_e + \alpha_i + \phi_f$, and we put

$$\varepsilon \leq \beta_i \ .$$

The linear objective function L, which is to be maximized, is just $L = \varepsilon$, what clearly maximizes the smallest angle $\beta_{\min}$ in the arc triangulation. There are precisely $3 \cdot (2n-h-2)$ inequalities and $3n-2h-3$ variables, if n is the total number of vertices, and h among them are situated on ∂D.

Sometimes the objective is to optimize not only the smallest angle, but rather to maximize lexicographically the sorted list of all arising angles, as is guaranteed by the Delaunay triangulation in the straight line case. This can be achieved by repeatedly solving the linear program above, keeping angles that have been optimized already as constants. By modifiying or adding constraints the results may be adapted to various needs, as avoiding angles larger than π or obtaining arc triangles 'as equilateral as possible'. We consider the flexibility of our simple approach as an important feature in practice.

[1]We have $d_{e'} = 0$ if e' is a line segment. However, we can keep $\phi_{e'}$ variable and bound it from above by some threshold $t > d_{e'}$.



(a) Delaunay triangulation (b) Arc triangulation

Figure 2: Flip-optimized arc triangulation starting from a Delaunay triangulation.

3 Flipping in Arc Triangles

The fact that every simple polygon can be triangulated with straight line segments is folklore. Again, a domain D with piecewise circular boundary need not admit *any* triangulation, even if circular arcs may be used. It is known that a linear number of Steiner points is required in the worst case to ensure an arc triangulation [1].

One of the arising questions is: Given the domain D and a (combinatorial) triangulation $\mathcal{T}_c$ in D (possibly with interior points), can $\mathcal{T}_c$ be realized by circular arcs? For deciding this, we can now utilize the linear program formulated in Section 2. A realizing arc triangulation exists if and only if the feasible region of the linear program is nonempty. As a particularly nice feature, this enables us to define flip operations in arc triangulations, as is described below. Consider some arc triangulation $\mathcal{A}$ in the domain D. Each interior arc $\widehat{pq}$ of $\mathcal{A}$ lies on the boundary of two arc triangles. Let r and s be the two vertices of these arc triangles different from p and q. Flipping $\widehat{pq}$ by definition means removing $\widehat{pq}$ from $\mathcal{A}$, establishing an arc between r and s combinatorially, and optimizing over the resulting triangulation. The new arc triangulation, if it exists, will contain a unique circular arc between r and s. In case of nonexistence, we declare the arc $\widehat{pq}$ as not flippable. Observe that an arc flip may change various circular arcs geometrically (by optimizing over their curvature), whereas only a single arc is exchanged combinatorially. An arc flip thus is a geometrically global operation which is combinatorially local.

Optimizing angles with arc flips is a powerful (though maybe costly) tool. We demonstrate the positive effect of sequences of such flips with Figures 2a and 2b. A significant improvement over the Delaunay triangulation becomes possible (in fact, the smallest angle is doubled in this example) by reducing the degree of a particular vertex, v. In general, we observe that small angles in a straight line triangulation stem from one of two reasons: (1) The geometry of the underlying domain D (plus its vertex set) forces slim triangles in the vicinity of ∂D. These 'boundary effects' can usually be mildened by mere geometric optimization of the corresponding arc triangulation. (2) Vertices of degree k naturally impose an upper bound of $\frac{2\pi}{k}$ on the smallest arising angle. This situation can be remedied only with combinatorial changes, and in contrast to the straight edge case, this is indeed possible for arc triangulations. (For straight edges, the combinatorics of the Delaunay triangulation is already optimal.)

4 Special Arc Triangles

An arc triangle ∇ is termed a *π-triangle* if the sum of its interior angles is π. Clearly, straight line triangles are π-triangles.

Property 1 *Let ∇ be some arc triangle. The following three properties are equivalent.*
(a) ∇ is a π-triangle.
(b) The three supporting circles of ∇ intersect in a common point exterior to ∇.
(c) ∇ is the image of a straight line triangle under a unique Möbius transformation.

Property 2 *Any π-triangle is contained in the circumcircle of its vertices.*

Proof. Omitted in this version. □

In view of the mentioned properties, it is worthwhile to study *π-triangulations*. Such triangulations will not always exist, depending on the boundary domain D, and in particular the sum of its inner angles, but they do, of course, if D is a simple polygon.

For the remainder of this section, let D be a simple polygon, and $\mathcal{T}$ be some straight line triangulation in D. The geometry of any arc triangulation $\mathcal{A}$ in D that is combinatorially equivalent to $\mathcal{T}$ is determined by the vector $\Phi(\mathcal{A})$ of deviation angles $\phi(a_i)$ for the interior arcs a_i of $\mathcal{A}$; see Section 2. Interpreting $\Phi(\mathcal{A})$ as a point in high dimensions, we can talk of the space of arc triangulations for $\mathcal{T}$. The next lemma is important in view of optimizing a given π-triangulation.

Lemma 1 *Let $\mathcal{T}$ have n vertices, h of which lie on the boundary of D. The dimension of the space of π-triangulations for $\mathcal{T}$ is $n - h$.*

Proof. Omitted in this version. □

Lemma 1 remains true if $\mathcal{T}$ is replaced by any π-triangulation of D. In practice, the input is most likely a straight line triangulation, which is to be optimized into a π-triangulation with maximum smallest angle. Figure 3 displays how a straight line triangulation is optimized into a π-triangulation. The change does not appear dramatic, but observe that the smallest angle (occurring at vertex v) almost doubles, from 9.7° to 19°. No arc flips have been applied. Table 1 shows experimental data for a larger input (500 random points, postprocessed to keep a certain interpoint distance as in realistic meshes). We see that the gain reduces for larger Delaunay meshes but is still significant, especially if the condition on the angle sum is relaxed from π to a small intervall around that value.

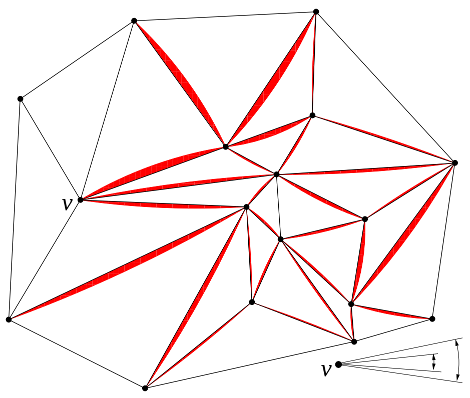


Figure 3: Straight line triangulation and its angle-maximized π-triangulation superimposed

angle sum	Delaunay min	min arc angle	gain
180°	18.03°	22.52°	25%
179° - 181°	-"-	22.92°	26%
175° - 185°	-"-	24.88°	38%
170° - 190°	-"-	27.53°	50%
160° - 200°	-"-	31.77°	72%

Table 1: Angle improvement in arc triangulations

5 Graph Drawing

Literature on drawing graphs nicely in the plane is large; see e.g. [5, 13]. Most algorithms take as input an abstract graph G and produce a layout of the vertices of G such that the resulting straight line (or orthogonal) drawing is aesthetically pleasing, and / or satisfies certain application criteria. On the theoretical side, bounds on the achievable angular resolution are known for various classes of graphs [7, 11], including planar graphs.

Results for curvilinear drawings of graphs are comparatively sparse. See, for example, [3, 9] and references therein, who give lower bounds and algorithms for drawing graphs on a grid with curved edges (including circular multiarcs), and [6] where a method based on physical simulation is proposed. To our knowledge, no algorithm has been given that draws a graph with (single) circular arcs under some optimization criterion. Here we actually consider a simpler setting, namely, for a given planar straight line embedding of a graph G, the problem of *redrawing* G with

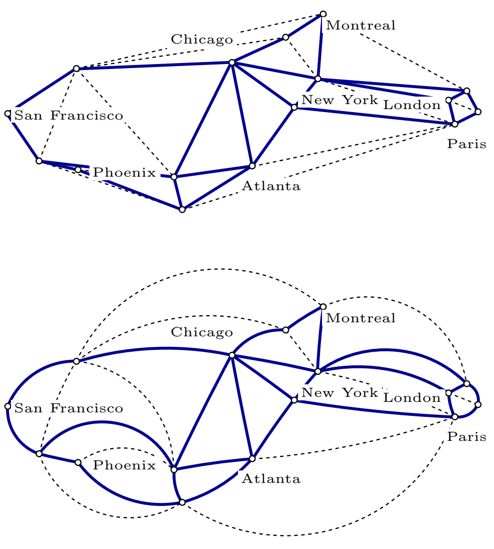


Figure 4: IP backbone graph, straight line and optimally redrawn.

curved edges in an optimal way. In a redrawing, the positions of the vertices are kept fixed. This may be a natural demand, for instance, in certain geographical applications. Let us describe how maximizing the smallest angle in a circular arc redrawing of G can be achieved very easily. It is tempting to apply the linear optimization method from Section 2 to G directly. This, however, bears the risk of arc overlaps getting out of control. The way out is to embed G in some triangulation $\mathcal{T}$ first, and treat respective sums of angles as single entities to be optimized. More precisely, for each angle ϱ in G, given by the concatenation of angles $\alpha_1, \ldots, \alpha_k$, $k \geq 1$, in $\mathcal{T}$ we use the constraint

$$\varepsilon \leq \sum_{i=1}^{k} \beta_i$$

with each β_i expressed by the corresponding straight line triangulation angle α_i and its two deviation variables ϕ_e and ϕ_f as in Section 2. Any completion to a triangulation of G will work (for instance, the constrained Delaunay triangulation [10, 4] of G), and the optimal solution does not depend on this choice. Also, the entire angle vector $\varrho_1, \ldots, \varrho_m$ for G can be optimized, in an iterative way as before. Additional restrictions may be posed, like $\varrho_j < \pi$ or $\varrho_j < \frac{\pi}{2}$, in order to preserve obtuse or sharp angles in G.

The adjacency graphs in Figure 4 exemplify the effect of our circular arc redrawing method. The results seem satisfactory, in spite of the fact that vertices are required not to move. Our results compare well to, e.g. [6], who use for optimization the additional freedom of placing vertices, though at a price of high computation cost. For our method, the number of vertices of the input graph is no limitation, as far as applications from graph drawing are concerned.

6 Future Work

Circular arc triangulations are a flexible and computationally controllable structure with potential impact but, so far, with lack of interest from computational geometry. Further open questions raised here are the convergence of the angle-increasing arc flipping process in Section 3, and an extension of the presented results to three dimensions with tetrahedral volumes with spherical faces. We will elaborate on the properties of such 3D primitives and their meshes in a forthcoming paper.

References

[1] O. Aichholzer, F. Aurenhammer, T. Hackl, B. Juettler, M. Oberneder, Z. Sir. Computational and structural advantages of circular boundary representation. Int'l J. Computational Geometry & Applications, to appear.

[2] M. Bern, D. Eppstein. Mesh generation and optimal triangulation. In: D.-Z.Du, F.Hwang (eds), Computing in Euclidean Geometry, Lecture Notes Series on Computing 4, World Scientific, 1995, 47-123.

[3] C.C. Cheng, C.A. Duncan, M.T. Goodrich, S.G. Kobourov. Drawing planar graphs with circular arcs. Proc. 7th Int. Symposium on Graph Drawing, 1999, Springer LNCS 1771, 2000, 117–126.

[4] L.P. Chew. Constrained Delaunay triangulations. Algorithmica 4 (1989), 97-108.

[5] G. Di Battista, P. Eades, R. Tamassia, I.G. Tollis. *Graph Drawing - Algorithms for the Visualization of Graphs.* Prentice Hall, 1999.

[6] B. Finkel, R. Tamassia. Curvilinear graph drawing using the force-directed method. Proc. 12th Int. Symposium on Graph Drawing, 2004, Springer LNCS 3383, 2004, 448-453.

[7] M. Formann, T. Hagerup, J. Haralambides, M. Kaufmann, F.T. Leighton, A. Symvonis, E. Welzl, G. Wöginger. Drawing graphs in the plane with high resolution. SIAM J. Computing 22 (1993), 1035 - 1052.

[8] S. Fortune. Voronoi diagrams and Delaunay triangulations. In: D.-Z.Du, F.Hwang (eds), Computing in Euclidean Geometry, Lecture Notes Series on Computing 4, World Scientific, 1995, 225-265.

[9] M.I. Goodrich, C.G. Wagner. A framework for drawing planar graphs with curves and polylines. J. Algorithms 37 (2000), 399–421.

[10] D.T. Lee, A.K. Lin. Generalized Delaunay triangulation for planar graphs. Discrete & Computational Geometry 1 (1986), 201-217.

[11] S. Malitz, A. Papakostas. On the angular resolution of planar graphs. Proc. 24th Ann. ACM Symp. on Theory of Computing, 1992, 527-538.

[12] J. Shewchuk. What is a good linear element? Interpolation, conditioning, and quality measures. Proc. 11th International Meshing Roundtable, 2002, 115-126.

[13] K. Sugiyama. *Graph Drawing and Applications for Software and Knowledge Engineers.* World Scientific, 2002.