Computing the Angularity Tolerance*

Mark de Berg † Henk Meijer ‡ Mark Overmars † Gordon Wilfong §

July 30, 1996

Abstract

In computational metrology one needs to compute whether an object satisfies specifications of shape within an acceptable tolerance. To this end positions on the object are measured, resulting in a collection of points in space. From this collection of points one wishes to extract information on flatness, roundness, etc. of the object. In this paper we study one particular feature of objects, the angularity. The angularity indicates how well a plane makes a specified angle with another plane. We study the problem in 2-dimensional space (where the planes become lines) and in 3-dimensional space. In 2-dimensional space the problem is equivalent to computing the smallest wedge of the a given angle that contains all the points. We give an $O(n^2 \log n)$ algorithm for this problem. In 3-dimensional space we study the more restricted problem where one of the planes is known (a datum plane). In this case the problem is equivalent to asking for the smallest width 3-dimensional strip that contains all the points and makes a given angle with the datum plane. We give an $O(n \log n)$ algorithm to solve this version. We also show that in the case of uncertainty in the measured points, upperbounds and lowerbounds on the width can be computed in similar time bounds.

1 Introduction

Manufactured objects are always approximations to some ideal object: parts that are supposed to be flat will not be perfectly flat, round parts will not be perfectly round, and so on. In many situations, however, it is important that the manufactured object is very close to the ideal object. In such cases the specification of an object includes a description of how far the manufactured object is allowed to deviate from the ideal one. The field of dimensional tolerancing [2] provides the language for this. Given a specification, one must test whether the manufactured object meets it, which is the area of study called computational metrology. The objects are often tested as follows. Suppose for simplicity that we want to manufacture a flat surface. First, a so-called Coordinate Measuring System (CMS) 'measures' the manufactured surface. The output of the CMS is a set of points in 3-dimensional space that are on the manufactured surface. The second step is to compute two parallel planes at minimum distance to each other that have all the measured points in between them. In other words, one wants to compute the width of the point set. The surface meets the requirement if the width is below the specified threshold. Computing the width of a point set can be done in $O(n \log n)$ time in the plane [3] and in $O(n^{3/2+\varepsilon})$ expected time in 3-dimensional space [1].

*This work was partially supported by the Netherlands Organisation for Scientific Research (N.W.O.) and by NSERC Canada. We also thank Bellairs Research Institute in Barbados for the use of their facilities.

†Department of Computer Science, Utrecht University, P.O.Box 80.089, 23508 TB Utrecht, the Netherlands.

‡Department of Computing and Information Science, Queen's University, Kingston, Ontario, Canada.

§Bell Labs, 600 Mountain Ave., Murray Hill, NJ 07974, U.S.A.

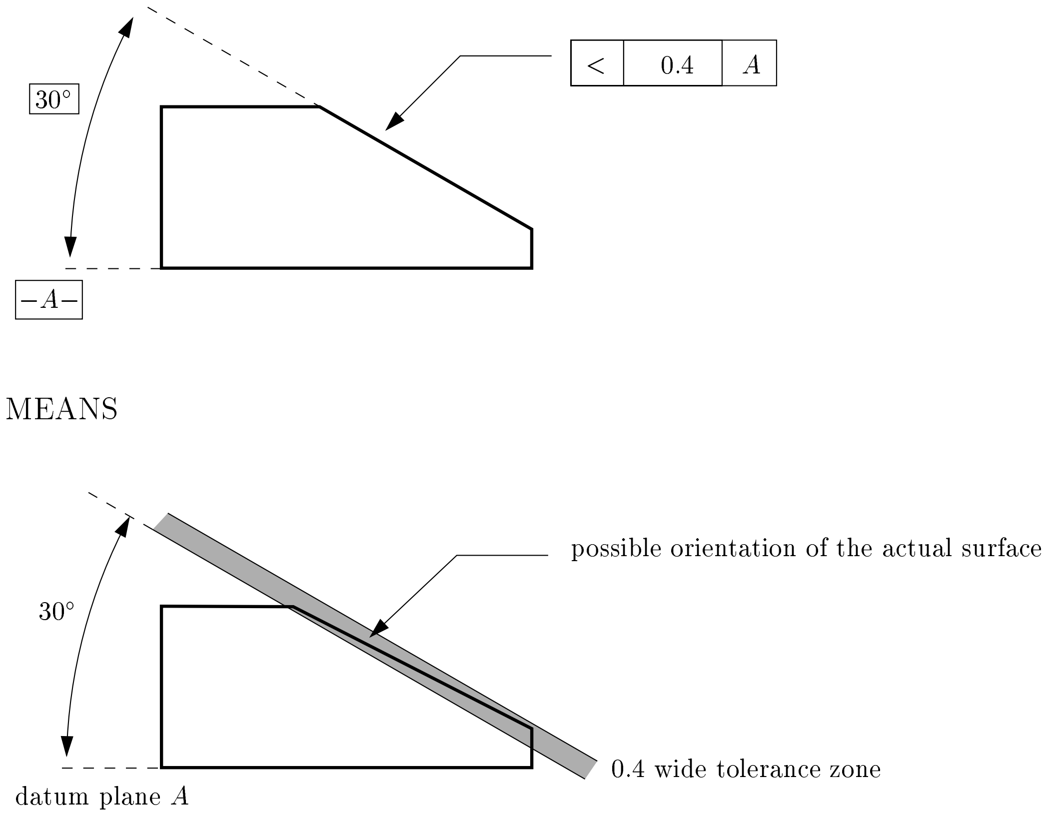

Figure 1: The pictorial definition of angularity tolerance.

We study another problem from computational metrology, which arises when one wants to manufacture an object with two flat surfaces that make a specified angle with each other. Testing whether the manufactured object meets the specifications leads to the *angularity problem*: given a set of points, compute a thinnest wedge whose legs make a given angle with each other and that contains all the points. We show that in the plane this problem can be solved in $O(n^2 \log n)$ time. In 3-dimensional space we study a simpler variant, where all the measured points come from one of the two surfaces; the other surface, the so-called *datum plane*, is assumed to be in a known orientation. In [2] this type of angular tolerance is indicated with the picture in Figure 1. The problem is now to find the thinnest 'sandwich' (that is, two parallel planes) that makes a given angle with the datum plane and contains all the points. In other words, we want to compute the width under the restriction that the planes make a given angle with the datum plane. We solve this problem in $O(n \log n)$ time. Both in the planar case and in the 3-dimensional case we also study variants where the points have uncertainty regions associated with them. These uncertainty regions occur in practise because the CMS is not completely precise but introduces some (small) error in the measurements.

2 Point sets in two dimensions

We start by studying the angularity problem in the plane. In the simplest version we are given a datum line, a set of n points (which are on one side of the line), and an angle θ. The problem then is to compute the thinnest strip (or, *sandwich*) that contains all the points and makes an angle θ with the datum line.

Definition 1 *A θ-sandwich of width δ is the closed area bounded by two parallel lines that make an angle θ with the datum plane. The width of the sandwich is the distance between the two lines.*

In the plane this simple version is not so interesting; it can easily be solved in linear time by computing the extrema of the point set in the direction perpendicular to θ. Therefore we concentrate on the case where the datum plane is not given. In this setting we are only given a set S of n points and an angle θ, and we want to compute the thinnest θ-wedge that contains all the points, where a θ-wedge is defined as follows:

Definition 2 *A θ-wedge of width δ is the closed area bounded by four directed half lines b_1, b_2, l_1 and l_2 such that*

- *b_1 is parallel to and to the right of b_2 and l_1 is parallel to and to left of l_2*
- *b_1 and l_1, as well as b_2 and l_2 have a common starting point*
- *the angle measured in counter clockwise direction between b_1 and l_1 is equal to θ*
- *the distance between b_1 and b_2 and between l_1 and l_2 is δ*

Figure 2 shows a wedge containing all points shown. The minimum δ such that there is a θ-wedge

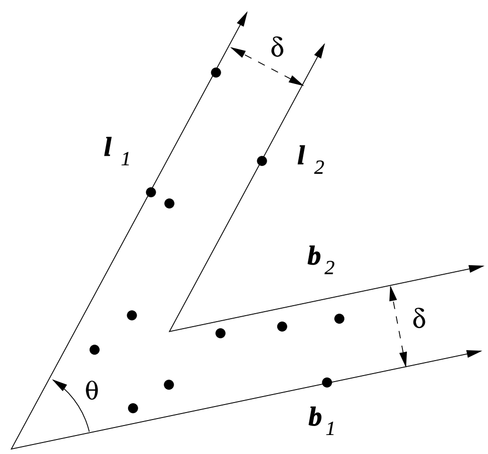

Figure 2: A θ-wedge of width δ.

of width δ containing S is called the *tolerance* of S (with respect to θ-wedges).

Toussaint and Ramaswami [6] have solved a simpler variant of the problem, where it is known in which of the two 'legs' of the wedge each input point lies: one is given an angle θ and two sets of points, each of which has to be enclosed in a strip such that the angle between the two strips is θ. Their algorithm runs in time $O(n \log n)$.

Let $W(\phi)$ be a θ-wedge of minimal width, such that the bisector of b_1 and l_1 has direction ϕ and the wedge contains S. If there is no point of S on b_1 we can move $W(\phi)$ along l_1 until at least one point of S is on b_1, while S remains contained in $W(\phi)$. So without loss of generality we may assume that there is at least one point of S on b_1 and, similarly, at least one on l_1. It is now easy to see that $W(\phi)$ is unique for each value of ϕ. Let $A(\phi)$ be the apex of $W(\phi)$. Let OC, the outer curve, be the collection of all points $A(\phi)$ for $0 \leq \phi \leq 2\pi$. Let $B(\phi)$ be the common starting point of b_2 and l_2 of wedge $W(\phi)$, and define the inner curve IC as the collection of all points $B(\phi)$. Our algorithm to compute the thinnest θ-wedge containing S starts by computing the curves OC and IC. The next two lemmas state how these curves look, and how much time we need to compute them.

Lemma 1 *The collection OC is a closed curve of piece-wise circular arcs, has a linear combinatorial complexity, and can be computed in $O(n \log n)$ time.*

Proof: Consider wedge $W(\phi)$. The points of S which lie on b_1 and l_1 are points of the convex hull of S. Suppose p_i lies on b_1 and p_j lies on l_1, as shown in Figure 3. The three points $A(\phi)$, p_i and p_j define a circle C, which passes through these three points. The circle is determined by the points p_i and p_j and the angle θ, so all wedges $W(\phi)$ for which p_i lies on b_1 and p_j on l_1 have their apex on this circle C. It is easy to see that this implies that the number of circular arcs of OC is linear in the size of the convex hull of S. Since the convex hull can be computed in $O(n \log n)$ time, OC can be determined in this time as well. ∎

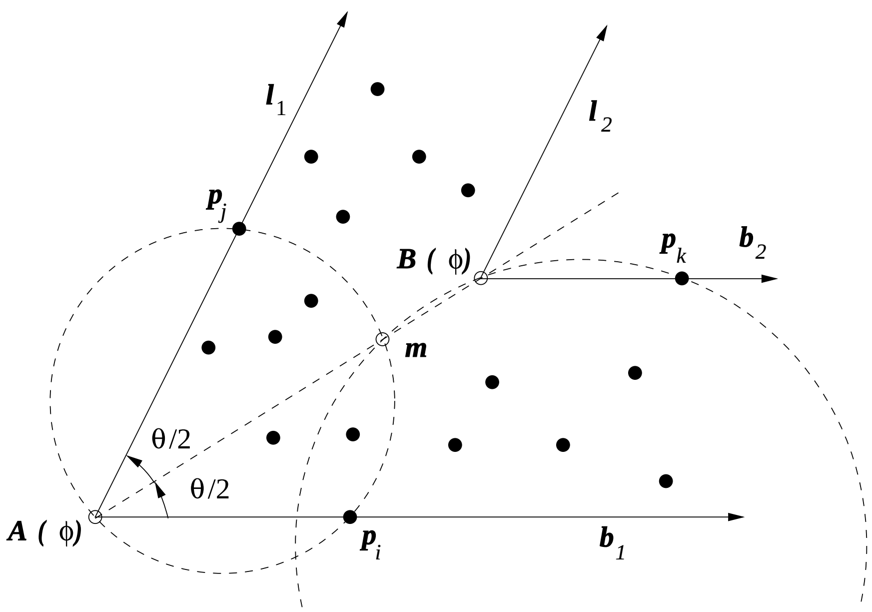

Figure 3: Outer and inner curve

Lemma 2 *Given an angle θ, the collection IC is a closed curve of piece-wise circular arcs, and has a combinatorial complexity of $O(n^2)$.*

Proof: Consider those values of ϕ for which the points $A(\phi)$ lie on the same circular arc of OC. Let p_i and p_j be the points on b_1 and l_1 respectively for these values of ϕ. So all wedges in this range have a bisector which passes through the point m, which lies on the circle through $A(\phi)$, p_i and p_j, halfway between p_i and p_j. See also Figure 3. Clearly for the wedge $W(\phi)$, there is at least one point of S on l_2 or b_2. Assume that point p_k is on b_2. If $B(\phi)$ is not between m and $A(\phi)$ then the angle between $(m, B(\phi))$ and $(B(\phi), p_k)$ is $\pi - \theta/2$. So m, $B(\phi)$ and p_k lie on a circle C, which is determined by the points m and p_k and the angle $\pi - \theta/2$. Therefore all wedges $W(\phi)$ in this range of values for ϕ for which p_k lies on b_2 have $B(\phi)$ on the circle C. If $B(\phi)$ is between m and $A(\phi)$, then the angle between $(B(\phi), m)$ and (m, p_k) is $\theta/2$. Since this angle is the complement of $\pi - \theta/2$, $B(\phi)$ lies on the same circle C. Therefore IC consists of piecewise circular arcs.

If one of the circular arcs of OC consists of a single point, then it can be shown that the corresponding IC is again a collection of circular arcs.

In both cases, for each circular arc of the outer curve OC, IC can have at most a linear number of circular arcs, so the lemma follows. ∎

Lemma 3 *Given an angle θ and a set of points S, the inner curve IC can be computed in $O(n^2 \log n)$ time.*

Proof: Let ϕ_1 and ϕ_2 be two directions such that for each wedge $W(\phi)$with $\phi_1 \leq \phi \leq \phi_2$, the apex $A(\phi)$ lies on the same circular arc of OC and the bisector passes through the point m. Let m_1 and m_2 be the two lines with directions ϕ_1 and ϕ_2 through the point m, as shown in Figure 4. Let BA be the area between the lines m_1 and m_2, i.e. the area where the bisectors of wedges within this range of values for ϕ can lie. For each point p in S we compute the two circles of radius r through m and p such that the length of the arc between m and p is θr. So for each point q on one of these circles, the angle between (m, q) and (p, q) is $\theta/2$ or $\pi - \theta/2$. Therefore for each ϕ, $B(\phi)$ has to lie on one of these circles. For each circle, we only keep its intersection with the area BA.

Let ϕ be a direction between ϕ_1 and ϕ_2 and let $C(p)$ be a circular arc in BA corresponding to a point p. The line with direction ϕ through the point m intersects $C(p)$. One of the points of intersection is the point m; let the other point of intersection be $c(p)$. If the line is tangent to $C(p)$, we define $c(p) = m$. Notice that $c(p)$ is the location of $B(\phi)$ of that wedge for which p lies on b_2 or l_2. So in order to find the wedge containing S for this direction ϕ, we have to compute $c(p_i)$ for all points p_i and find the one that has maximum distance from $A(\phi)$. Computing IC in this range of values for ϕ corresponds to computing the outer envelope of functions which intersects each other at most once. We can compute IC in this range of values for ϕ by a divide and conquer algorithm in $O(n \log n)$ time. Therefore the complete curve IC can be computed in $O(n^2 \log n)$ time. ∎

After computing OC and IC it is easy to compute the thinnest wedge: We split the range $[0 : 2\pi]$ of possible orientations of the wedge into subranges where both the outer curve and the inner curve are attained by a single (piece of a) circular arc. The previous two lemmas imply that there are $O(n^2)$ subranges. (In fact, the subranges correspond exactly to the arcs of IC.) For each subrange we can then compute the thinnest wedge in constant time. This leads to the following result.

Theorem 1 *Given an angle θ and a set of n points S, the thinnest θ-wedge containing S can be found in $O(n^2 \log n)$ time.*

The running time of our algorithm is dominated by the time to compute IC. One might hope to improve this by showing a better bound on the complexity of IC. The next theorem shows that this is not possible.

Lemma 4 *The worst-case complexity of IC is $\Theta(n^2)$.*

Proof: Let n be an arbitrary large integer divisible by 4 and greater than 7, say. Let θ be equal to π/n. Construct a regular n-gon and draw an axis of symmetry which does not intersect a vertex of the n-gon. Draw points of S on the locations of the $n/2$ vertices of the n-gon closest to the axis. Figure 5 shows an example with $n = 16$. Place the remaining $n/2$ points on the axis of symmetry

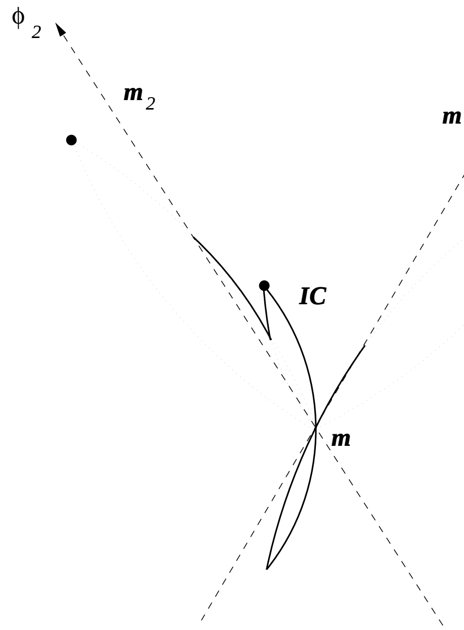

Figure 4: Section of the inner curve IC

and close to the centre of the n-gon, in fact much closer than shown in Figure 5. We call these points the inner points. It can be shown that the line segments b_1 and l_1 will either contain an outer point plus a neighbour of its opposite point, such as p_{i-1} and p_j in the figure, or they contain two opposite outer points of S. In the first case, the point m is between the centre of the n-gon and $A(\phi)$. If $A(\phi)$ varies over its circular arc in OC the bisectors of the corresponding wedges pass all inner points if they are sufficiently close to the centre, so IC consists of a n of circular arcs, two for each inner point. Similarly, if b_1 and l_1 contain two opposite outer points, m is not between the centre of the n-gon and $A(\phi)$, and again IC has a linear number of arcs if $A(\phi)$ varies over its circular arc in OC. Therefore IC has a quadratic combinatorial complexity. ∎

Using only the curve OC we can solve the more restricted problem of covering a set S which is partitioned into two subsets, such that each arm of the wedge contains one of the subsets, yielding the same result as in [6].

Theorem 2 *If the set of points S is partitioned into sets S_1 and S_2 then the thinnest wedge of angle θ containing S with S_1 between b_1 and b_2 and S_2 between l_1 and l_2 can be computed in $O(n \log n)$ time.*

Proof: We define ϕ and $W(\phi)$ as before except that S_1 and S_2 lie between b_1 and b_2 and between l_1 and l_2 respectively. For each direction ϕ we compute $d_1(\phi)$, which we define as the maximal distance of a point of S_1 to b_1. Similarly we compute $d_2(\phi)$, the maximal distance of a point of S_2 to b_2. So the width of $W(\phi)$ is $max(d_1(\phi), d_2(\phi))$.

Therefore the only points to be considered in the computation of $d_1(\phi)$ and $d_2(\phi)$ are points of $CH(S_1)$ and $CH(S_2)$, the convex hulls of S_1 and S_2 respectively. We examine the arcs in OC in clockwise order. For one of these arcs, we compute the thinnest $W(\phi)$ for all wedges with $A(\phi)$ on this arc in a time which is linear in the size of $CH(S_1)$ and $CH(S_2)$. For each subsequent arc in OC, we now compute the thinnest wedge with apex on this arc by walking clockwise around

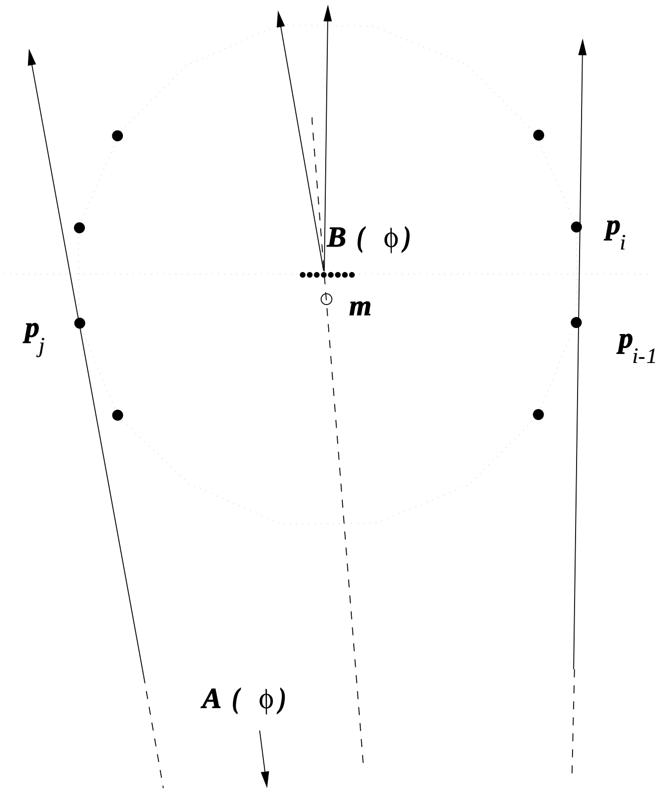

Figure 5: Inner curve of size $\Theta(n^2)$

$CH(S_1)$ and $CH(S_2)$. So once we have computed $CH(S_1)$ and $CH(S_2)$, the thinnest wedge can be found in linear time. Therefore the problem is solvable in $O(n \log n)$ time. ∎

3 Uncertainty regions in two dimensions

In computational metrology the sample points normally are not exact but come with some uncertainty: rather than a set of points, we get a set U of uncertainty regions $\{u_1, ..., u_n\}$. For each region u_i there is a point $p_i \in u_i$ that lies on the surface of the manufactured object, but due to the inaccuracy in the measuring process the point p_i is not known. In this case one would like to compute upper and lower bounds on the tolerance.

To illustrate the definitions, we first look at the simple version of the problem, where we are given a datum line, a set U of uncertainty regions, and an angle θ. For a set S of points, define $\delta(\theta, S)$ to be the tolerance of S, that is, $\delta(\theta, S)$ is the width of the thinnest θ-sandwich (strip) that contains S. An upper bound on the tolerance of any set $S = \{p_1, \ldots, p_n\}$ of points within the uncertainty regions of $U = \{u_1, ..., u_n\}$ is given by the quantity

$$\max\{\delta(\theta, S) | p_i \in u_i \text{ for } 1 \leq i \leq n\}. \quad (1)$$

Unfortunately this quantity is hard to compute. Therefore we compute a more conservative upper bound, $\delta_{\max}(\theta, U)$, defined as follows:

$$\delta_{\max}(\theta, U) = \text{minimum width } \theta\text{-sandwich containing all uncertainty regions in } U.$$

The value $\delta_{\max}(\theta, U)$ is called the maximum tolerance of U. Notice that $\delta_{\max}(\theta, U)$ is also the width of the thinnest θ-sandwich that is guaranteed to contain all points of any set $S = \{p_1, \ldots, p_n\}$ with $p_i \in u_i$. At first glance it might seem that $\delta_{\max}(\theta, U)$ is the same as the upper bound given by equation (1), but this is not true. We get a trivial example by taking a set U consisting of only one region, say the unit circle. In this case we have $\delta_{\max}(\theta, U) = 2$ and $\max \delta(\theta, S) = 0$. But also for a larger number of uncertainty regions $\delta_{\max}(\theta, U)$ can be greater than $\max \delta(\theta, S)$. Consider the example with two regions shown in Figure 6. In the example $\theta = \pi/3$ and U consists of two unit circles. In this example $\delta_{\max}(\theta, U) = 2 + \sqrt{3}$. However, $\delta(\theta, S) \leq 2\sqrt{3}$ for all choices of S. (If we place the points of S where the thinnest sandwich for U touches the two circles, then we can obtain a thinner θ-sandwich by 'flipping' the sandwich, so that the angle with the datum line (here the x-axis) is no longer given by the angle with the positive x-axis but with the negative x-axis. In the planar case one could argue that the angle of the flipped sandwich is not θ but $\pi - \theta$. In 3-dimensional space, however, a similar example applies, and there it is natural to allow 'rotating' the sandwich while keeping the angle with the datum plane fixed.)

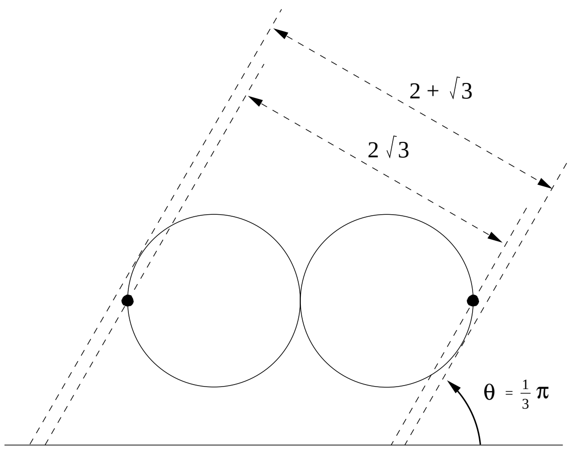

Figure 6: Two regions of uncertainty

When we define a lower bound on the tolerance, we do not get these problems; we define

$$\delta_{\min}(\theta, U) = \min\{\delta(\theta, S) | p_i \in u_i \text{ for } 1 \leq i \leq n\}.$$

The value $\delta_{\min}(\theta, U)$ is called the minimum tolerance of U. The minimum tolerance is the same as the width of the thinnest sandwich which contains at least one point from each uncertainty region u_i. In order to compute sandwiches containing uncertainty regions, we make the assumption that given a direction ϕ and an uncertainty region u_i, we can compute the two tangents of u_i with direction ϕ in constant time. Computing the maximum and minimum tolerance of a given set of uncertainty regions, when we are given a datum line and an angle θ is trivial to do in linear time.

Now consider the case where no datum line is given. The definitions of maximum and minimum tolerance readily carry over.

In order to compute the thinnest wedge containing a set of uncertainty regions, we can proceed as in the previous section. The combinatorial complexity of the curves OC and IC depends on the shape of the uncertainty regions. For example if all regions are equal size circles, the curve IC does not consist of circular arcs, but it has at most a quadratic complexity. In this case, using the same approach as above, one can prove the following result:

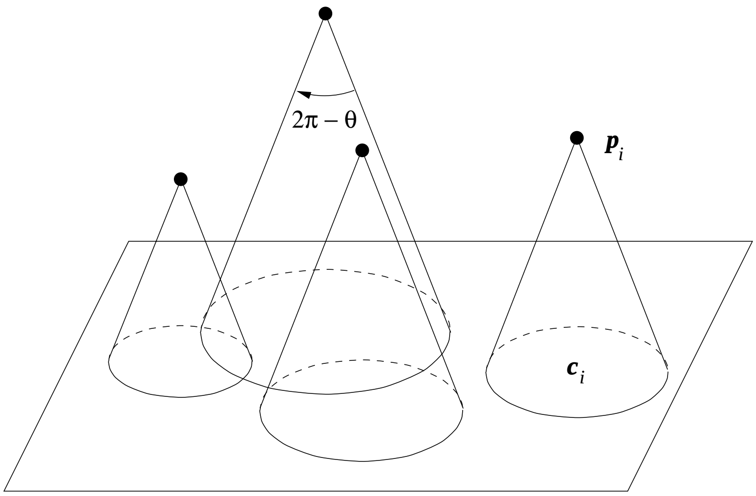

Figure 7: Turning the points p_i into circles c_i.

Theorem 3 *Given a set of n uncertainty regions U consisting of equal size circles and an angle θ, the value $\delta_{\max}(\theta, U)$ can be found in $O(n^2 \log n)$ time.*

It is an open problem to determine which other shapes of uncertainty regions permit an $O(n^2 \log n)$ algorithm.

4 Point sets in three dimensions

In the 3-dimensional setting we only study the simple variant of the angularity problem, where a datum plane is given. We assume without loss of generality that the datum plane is the xy-plane. The set of points, which we assume to lie above the x-plane, is denoted by S. We want to compute the thinnest θ-sandwich containing all points in S, where θ is a given angle. The definition of a θ-sandwich in 3-space is a generalisation of the definition of a sandwich in 2-space.

Definition 3 *A θ-sandwich of width δ is the closed area bounded by two parallel planes that make an angle θ with the datum plane. The width of the sandwich is the distance between the two planes. We denote the plane bounding the sandwich from above by h_1 and the plane bounding it from below by h_2.*

To find the pair h_1 and h_2 bounding the thinnest sandwich, we transform the problem into a 2-dimensional problem as follows. (We could also work directly in 3-space, but we feel that the transformation makes the algorithm easier to understand, especially in the case of uncertainty regions, which is studied later.) For a point $p_i \in S$ let C_i be the cone pointing upwards with apex p_i (thus, p_i is the highest point of the cone) and apex angle $\pi - 2\theta$, that is, the sides make an angle of θ with the xy-plane as shown in Figure 7. Now p_i lies below or on h_1 if and only if C_i lies below or on h_1. Similarly, p_i lies above h_2 if and only if C_i does not lie completely below or on h_2. Each cone C_i intersects the xy-plane in a circle c_i. The plane h_1 intersects the xy-plane in a line l_1, and h_2 intersects it in a line l_2 parallel to l_1. We direct the lines l_1 and l_2 so that l_2 is to the left of l_1. The cone C_i lies below h_1 if and only if the circle c_i lies to the left of l_1. Similarly C_i lies below h_2 if and only if circle c_i is to the left of l_2. This leads to the following 2-dimensional reformulation of the problem: Given a set of circles $\{c_1, ..., c_n\}$, determine two parallel directed lines l_1 and l_2 such that

i) All circles lie to the left of or on l_1.

ii) No circle completely lies to the left of l_2.

iii) Among all pairs of lines that satisfy i) and ii) the distance between l_1 and l_2 is minimal.

It is easy to verify that this is indeed the same problem. Figure 8 shows an example of a solution for such a 2-dimensional problem. We can now solve the problem as follows. Consider a valid pair

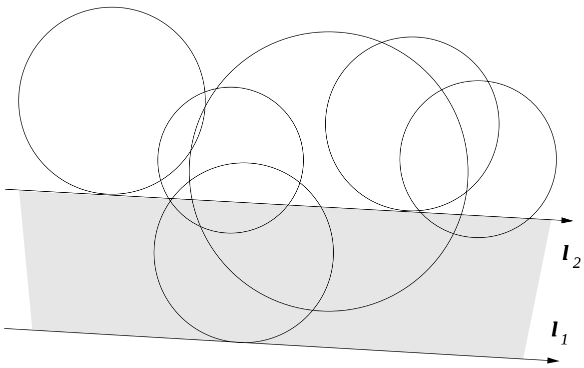

Figure 8: Solution to the planar problem.

of lines of minimum width, where we fix the slope of the lines to be, say, zero. For each region, take the lowest point on its boundary. Now the lines l_1 and l_2 go through the lowest and the highest point, respectively, of all such lowest points. When we start increasing the slope of the lines from zero to 2π, then the extreme points defining l_1 and l_2 move along the boundaries of the regions on which they lie. At some point they will switch from one region to another one. Which two regions define the two extreme points for a given slope ϕ can be determined by computing suitably defined lower and upper envelopes. The thinnest sandwich is then determined by the minimum distance between these two envelopes. A more detailed description is given in the proof below.

Theorem 4 *Given a set of points S in 3-space and an angle θ, the minimum width θ-sandwich that contains S can be found in $O(n \log n)$ time.*

Proof: Let circle c_i have centre (x_i, y_i) and radius r_i. Let ϕ be an angle between 0 and 2π. Let m_1 and m_2 be two directed parallel lines with normal vectors equal to $(\cos\phi, \sin\phi)$, and such that all circles c_i lie to the left of or on m_1, no circle completely lies to the left of m_2 and among all such pairs, the lines m_1 and m_2 have minimal distance.

A directed line l with normal vector equal to $(\cos\phi, \sin\phi)$, tangent to c_i such that c_i lies to the left of l, has distance d_i to the origin, where $d_i = x_i \cos\phi + y_i \sin\phi + r_i$, as shown in Figure 9. Notice that this defines d_i to be negative if the origin is to the right of the line l. Let $upper(\phi)$ and $lower(\phi)$ be defined by

$$upper(\phi) = \max_{1 \le i \le n} (x_i \cos\phi + y_i \sin\phi + r_i)$$

$$lower(\phi) = \min_{1 \le i \le n} (x_i \cos\phi + y_i \sin\phi + r_i)$$

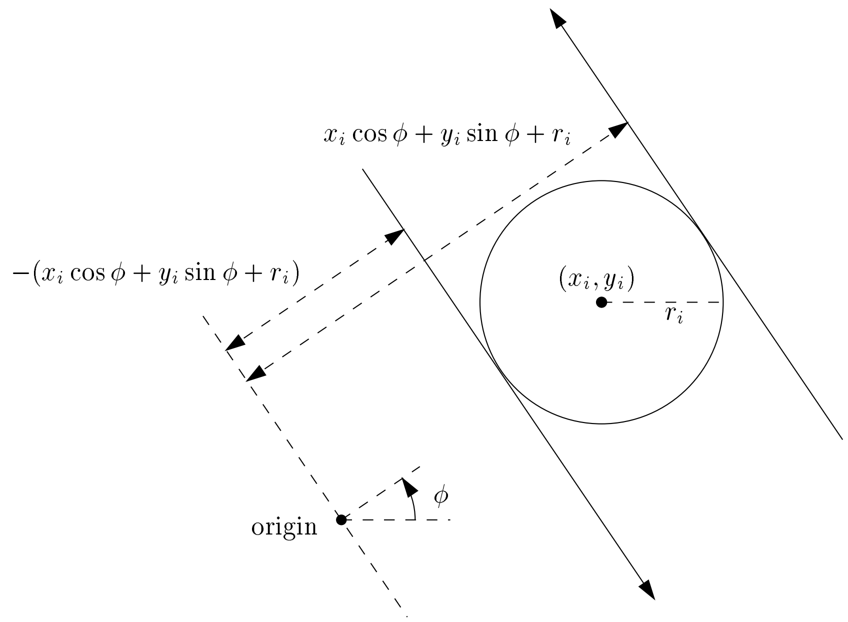

Figure 9: Distance to directed line tangent to a circle

So the distance between m_1 and m_2 is $upper(\phi) - lower(\phi)$, which is a non-negative value independent of the location of the origin. Therefore the lines l_1 and l_2 can be found by determining ϕ for which $upper(\phi) - lower(\phi)$ is minimised. Functions of the form $f_i(\phi) = x_i \cos\phi + y_i \sin\phi + r_i$ intersect each other at most twice between 0 and 2π. The intersections of $f_i(\phi)$ and $f_j(\phi)$ correspond to the common outer tangents of c_i and c_j and can also be computed by finding these tangents. (Hence, we never have to compute intersections between the functions directly, which is important for obtaining a fast and robust implementation.) From this we conclude that the functions $upper(\phi)$ and $lower(\phi)$ have a linear combinatorial complexity [4]. Using a divide-and-conquer algorithm, we can compute the functions $upper(\phi)$ and $lower(\phi)$ in $O(n \log n)$ time. By traversing these functions simultaneously from $\phi = 0$ to $\phi = 2\pi$ we can determine the value ϕ for which the difference is minimised in linear time. So the problem of finding the minimal width strip with an angle θ containing all points in S can be solved in $O(n \log n)$ time. ∎

5 Uncertainty regions in three dimensions

We define the maximum and minimum tolerance of a set U of uncertainty regions with respect to θ-sandwiches similar to the planar case—see Section 3. We first show how the thinnest θ-sandwich can be computed that contains a set of uncertainty regions U. As before we can transform the 3-dimensional problem into a 2-dimensional problem. Let P be a plane that has an angle θ with the $xy-$plane (which is again assumed to be the datum plane) and is tangent to region u_i, such that u_i is below P. We define the generalised cone C_i as the intersection of the half spaces below all such planes P. The intersection of C_i and the $xy-$plane is the region c_i. For example, if u_i is a sphere with positive z-coordinates, then C_i is an upwards pointing cone and c_i is a circle, as shown in Figure 10. Similarly, let Q be a plane which has an angle θ with the $xy-$plane and is tangent to u_i, such that u_i is above Q. The generalised cone D_i is the intersection of the half spaces

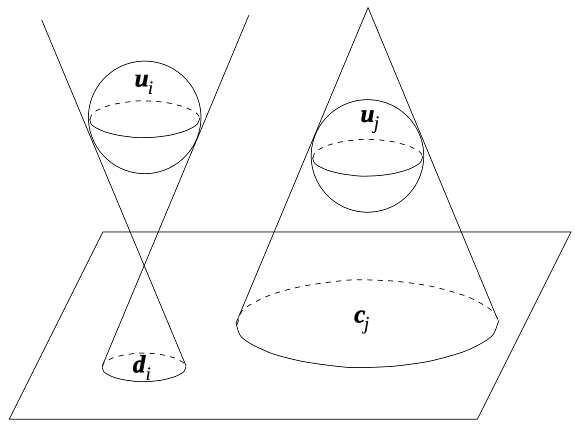

Figure 10: Turning 3-dimensional regions into 2-dimensional convex objects

below all such planes Q. The intersection of D_i and the $xy-$plane is the region d_i. So for all i we have $d_i \subset c_i$. Without loss of generality we can assume that all regions u_i have sufficiently large z-coordinates so that none of the regions d_i is empty.

Notice that, independent of the shape of u_i, the uncertainty regions c_i and d_i in the xy-plane are convex and have exactly one tangent through each point on their boundaries. We assume that the uncertainty regions c_i and d_i satisfy the following properties.

- Given a direction ϕ in the xy-plane and a region we can compute the two tangents of the region with direction ϕ in constant time.
- Given two non-intersecting regions, their common inner and outer tangents can be computed in constant time.
- The boundaries of two intersecting regions intersect each other at most k times where k is a constant and their common outer tangents can be computed in $O(1)$ time.
- For any connected part c of the boundary of a region c_i and any connected part d of the boundary of a region d_j we can find in constant time two parallel tangents through a point of c and d respectively which have minimal distance to each other.

Any two parallel planes that have an angle θ with the xy-plane and that are guaranteed to contain all points from U correspond to two directed parallel lines m_1 and m_2, such that the following holds:

i) All regions c_i lie to the left of or on m_1,

ii) No region d_i is completely to the left of m_2.

We can now find the thinnest θ-sandwich that contains all regions in U, that is, the maximum tolerance of U, with an algorithm that is similar to the algorithm we used for points. Let $\lambda_k(n)$ be the maximal length of an (n, k) Davenport-Schinzel sequence [4]. For example $\lambda_2(n) = 2n$-1 and $\lambda_k(n)$ is nearly linear for constant values of k.

Theorem 5 *Given a set of n uncertainty regions U and an angle θ, the maximum tolerance of U with respect to θ-sandwiches can be found in $O(\lambda_k(n) \log n)$ time.*

Proof: We proceed as we did when computing the thinnest sandwich containing a set of points in 3 dimensions. Let $f_i(\phi)$ be the distance of the origin to the directed line l, which is tangent to c_i, has c_i to its left and has the normal vector $(\cos\phi, \sin\phi)$. As before $f_i(\phi)$ is defined to be negative if the origin of the xy-plane is to the right of line l. Let $g_i(\phi)$ be the distance of the origin to the directed line l, which is tangent to d_i, has d_i to its left and has the normal vector $(\cos\phi, \sin\phi)$. Then we define

$$upper(\phi) = \max_{1 \le i \le n} f_i(\phi)$$

$$lower(\phi) = \min_{1 \le i \le n} g_i(\phi)$$

Let l_1 and l_2 be two lines who among all pairs of lines m_1 and m_2 that satisfy conditions i) and ii) above have the minimal distance from each other. In order to find l_1 and l_2, we have to find ϕ which minimises $upper(\phi) - lower(\phi)$. The combinatorial complexity of $upper(\phi)$ and $lower(\phi)$, which are the upper and lower envelopes of the functions $f_i(\phi)$ and $g_i(\phi)$ respectively, depend on the number of intersections of these functions. The number of times that $f_i(\phi)$ intersects $f_j(\phi)$ is equal to the number of common outer tangents of c_i and c_j. This is at most equal to the maximal number of intersections of the boundaries of c_i and c_j, which is assumed to be k. The points of intersection can be found by computing these common outer tangents.

Similarly, the number of times that $g_i(\phi)$ intersects $g_j(\phi)$ is equal to the number of common outer tangents of d_i and d_j. So we can find the intersections of $f_i(\phi)$ and $f_j(\phi)$ and of $g_i(\phi)$ and $g_j(\phi)$ for any i and j in constant time.

The combinatorial complexity of $upper(\phi)$ is $O(\lambda_k(n))$, which follows immediately from the results on upper envelopes in [4]. Similarly the combinatorial complexity of $lower(\phi)$ is $O(\lambda_k(n))$. We can find $upper(\phi)$ and $lower(\phi)$ in $O(\lambda_k(n) \log n)$ time by a divide-and-conquer algorithm. Since we assumed that we can compute the minimum of $f_i(\phi) - g_j(\phi)$ for any i and j and on any interval of $[0, 2\pi]$ in constant time, we can compute the value of ϕ for which $upper(\phi) - lower(\phi)$ is minimised in an additional $O(\lambda_k(n))$ time. ▮

A similar approach can be used to compute the minimum tolerance of U. In this case we need to find two parallel planes with angle θ such that there is at least one point in each region u_i between or on the two planes. This problem can also be transformed into an equivalent problem in 2 dimensions. Using the notation introduced for the maximum angular tolerance, this problem is as follows: Determine two parallel directed lines l_1 and l_2 such that

i) All regions d_i lie to the left of or on l_1,

ii) No region c_i is completely to the left of l_2.

iii) Among all pairs of lines that satisfy i) and ii) the distance between l_1 and l_2 is minimal.

Therefore the problem of computing the minimum angular tolerance has the same time complexity as the problem of computing the maximum angular tolerance.

For example, when the uncertainty regions u_i are spheres the value of k is 2, resulting in a time bound of $O(n \log n)$. The same result holds when the regions are homothetic copies of each other, that is, they have the same shape and orientation (but not necessarily the same size). For arbitrary constant complexity regions the bound becomes only slightly worse.

6 Conclusions

In this paper we studied the angularity tolerance problem in 2- and 3-dimensional space. We modelled them as sandwich problems and gave efficient algorithms, both in the case of exact measures and in the case of uncertainty regions. The problem of finding a thinnest wedge was only solved for the 2-dimensional case. The 3-dimensional case remains open. Computational metrology also involves many other geometric questions for which solutions need to be found. We refer to [7] for an overview.

7 Acknowledgements

We thank Godfried Toussaint for organising the 1995 Workshop on Computational Metrology at the McGill University's Bellairs Research Institute, where this work was started. We also thank the participants of this workshop for their help.

References

[1] P.K. Agarwal and M. Sharir. Efficient randomized algorithms for some geometric optimization problems. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 326–335, 1995.

[2] ASME Y14.5M-1994. *Dimensioning and Tolerancing.* The American Society of Mechanical Engineers, New York, 1994.

[3] M.E. Houle and G.T. Toussaint. Computing the width of a set. *IEEE Trans. Pattern Anal. Mach. Intell.* 5:761–765, 1988.

[4] M. Sharir and P.K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications.* Cambridge University Press, 1995.

[5] T.C. Schermer, C. Yap, Probing for near-centers and approximate relative roundness. ASME Workshop on Tolerancing & Metrology (invited talk), University of North Carolina at Charlotte, June 21-23, 1995.

[6] G. Toussaint and S. Ramaswami, Personal Communication, 1995.

[7] C. Yap, Exact Computational Geometry and Tolerancing Metrology. In *Snapshots of Computational and Discrete Geometry, Vol.3.* (eds. D.Avis and J.Bose), McGill School of Comp.Sci, Tech.Rep. No.SOCS-94.50, 1994.