SERIE B — INFORMATIK

Smallest Enclosing Ellipses – Fast and Exact *

Bernd Gärtner†
Sven Schönherr‡

B 97-03
May 1997

Abstract

The problem of finding the smallest enclosing ellipsoid of an n-point set P in d-space is an instance of convex programming and can be solved by general methods in time $O(n)$ if the dimension is fixed. The problem-specific parts of these methods are encapsulated in *primitive operations* that deal with subproblems of constant size. We derive explicit formulae for the primitive operations of Welzl's randomized method [22] in dimension $d = 2$. Compared to previous ones, these formulae are simpler and faster to evaluate, and they only contain rational expressions, allowing for an exact solution.

* supported by the ESPRIT IV LTR Project No. 21957 (CGAL), a preliminary version appeared in *Proc. 13th Annu. ACM Symp. on Computational Geometry (SoCG), 1997* [8]
† Institut für Theoretische Informatik, ETH Zürich, Haldeneggsteig 4, CH-8092 Zürich, Switzerland, e-mail: `gaertner@inf.ethz.ch`
‡ Institut für Informatik, Freie Universität Berlin, Takustr. 9, D-14195 Berlin, Germany, e-mail: `sven@inf.fu-berlin.de`

1 Introduction

The unique ellipsoid of smallest volume enclosing a compact set P in d-space (also known as the *Löwner-John ellipsoid* of P [10]) has appealing mathematical properties which make it theoretically interesting and practically useful. In typical applications, a complicated body needs to be covered by a simple one of similar shape and volume, in order to simplify certain tests. For convex bodies (e.g. the convex hull of a finite point set), the smallest enclosing ellipsoid – unlike the isothetic bounding box or the smallest enclosing sphere – guarantees a volume approximation ratio that is independent of the shape of the covered body. This follows from the following property of it, first proved by John (who also established existence and uniqueness) [10]: if the smallest enclosing ellipsoid of a compact convex body K is scaled about its center with factor $1/d$, the resulting ellipsoid lies completely inside K. Let us mention three concrete applications.

Ray tracing. Given a scene of objects in 3-space and a ray, find the first object hit by the ray. This problem occurs in computer graphics, when the scene is rendered in presence of various light sources. Since many rays need to be processed, the query has to be answered fast. In order to test whether a given object is hit by a ray, we can first test with a bounding volume (which we have precomputed) – if the ray misses it, it misses the object as well [7, section 15.10.2]. As bounding volumes, boxes, spheres but also ellipsoids [3] are useful, depending on the kind of objects.

Motion planning. In a similar spirit, bounding volumes are applied in robotics, when a collision-free motion of a robot among a set of obstacles is sought. After enclosing the robot and/or the obstacles by simple shapes, the problem becomes easier, and if a valid motion is found in the simplified environment, this motion is also valid in the original setting. It is clear that this heuristic is the more successful, the tighter the bounding volumes approximate the objects [4].

Statistics. In a different way, the smallest enclosing ellipsoid is applied in statistics. Given a cloud of measure points in d-space, one wants to identify and peel off 'outliers', often repeatedly. One heuristic peels off the vertices of the convex hull [2]. A finer peeling is obtained by choosing the boundary points of the smallest enclosing ellipsoid [18], whose number typically depends only on d.

1.1 Previous Work

Several algorithms for computing the smallest enclosing ellipsoid of an n-point set in d-space have been proposed. On the one hand, there are iterative methods which employ standard optimization techniques (such as gradient descent), adapted to the problem [21, 17]. These algorithms usually work on a dual problem, known as *D-optimal design* [20]. On the other hand, there are finite methods which find the desired ellipsoid within a bounded number of steps. For fixed d, the algorithm of Post [15] has complexity $O(n^2)$. An optimal deterministic $O(n)$ algorithm has been given by Dyer [6], randomized $O(n)$ methods are due to Adler and Shamir [1] and Welzl [22]. Since the problem is *LP-type* in the sense of [12], generic algorithms for this class of problems can be applied as well, see [9]. In any case, the runtime dependence on d is exponential. A method for the case $d = 2$, without time analysis, has been developed by Silverman and Titterington [18].

All these finite methods have the property that actual work is done only for problem instances whose size is bounded by a function in d. Assuming that d is constant, such instances can be solved in constant time. However, as far as explicit formulae for these *primitive operations* have been given – which is the case only for $d = 2$ – they are quite complicated and rely on solving third-degree polynomials [18, 14, 16]. This makes them expensive to evaluate and only leads to approximate solutions, unless specialized number types allowing exact manipulations of expressions involving roots (like LEDA's number type `real` [13]) are used.

1.2 Our Contribution

The goal of this paper is to show that in case of Welzl's algorithm for $d = 2$, the primitive operations can be implemented in rational arithmetic. This means, they can be performed exactly, if multiple precision integers (like LEDA's number type `integer` [13] or the GNU Multiple Precision Arithmetic Library[1]) are used. Even if the computations are done in floating point arithmetic, the simple rational expressions we get guarantee that the primitives are easier to code and more efficient to evaluate than in previous methods.

In the two-dimensional case we treat here, the constant-size problems involve smallest ellipses defined by up to 5 points, where the difficult case arises when the ellipse is defined by 4 points. As we show below, even if the points have rational coordinates, the ellipse will typically have not, so in order to stay with rational expressions, an explicit evaluation of the ellipse has to be avoided.

The main problem now is to perform the crucial primitive of Welzl's method, namely to test whether a point lies inside a given ellipse, where this ellipse may have irrational coordinates and is therefore not explicitly given.

Below we reduce this *in-ellipse test* to a sign evaluation of a certain derivative, and this leads to an elegant and efficient method whose computational primitives are in-ellipse tests over rational ellipses and evaluations of derivatives at rational values. Plugged into Welzl's algorithm, this solves the whole problem of computing smallest enclosing ellipses in rational arithmetic.

2 Smallest Enclosing Ellipsoids

Let us briefly review Welzl's randomized algorithm for computing the smallest enclosing ellipsoid of an n-point set in d-space [22]. The algorithm is very simple and achieves an optimal expected runtime of $O(n)$ if d is constant.

Given a point $c \in \mathbb{R}^d$ and a symmetric, positive definite[2] matrix $A \in \mathbb{R}^{d \times d}$, the set of

[1] available by anonymous ftp from from `prep.ai.mit.edu`. The file name is `/pub/gnu/gmp-M.N.tar.gz`

[2] i.e. $x^T A x > 0$ for $x \neq 0$

points $p \in \mathbb{R}^d$ satisfying

$$(p - c)^T A (p - c) = 1 \tag{1}$$

defines an *ellipsoid* with center c. The function $f(p) = (p - c)^T A (p - c)$ is called the *ellipsoid function*, the set $E = \{p \in \mathbb{R}^d \mid f(p) \leq 1\}$ is the *ellipsoid body.* Given a point set $P = \{p_1, \dots, p_n\} \subseteq \mathbb{R}^d$, we are interested in the ellipsoid body of smallest volume containing P. Identifying the body with its generating ellipsoid, we call this the *smallest enclosing ellipsoid* of P, $\textsc{SmEll}(P)$. (If the affine hull of P is not equal to $\mathbb{R}^d$, $\textsc{SmEll}(P)$ is a lower-dimensional ellipsoid 'living' in the affine hull).

The idea of Welzl's algorithm for computing $\textsc{SmEll}(P)$ is as follows: if P is empty, $\textsc{SmEll}(P)$ is the empty set by definition. If not, choose a point $q \in P$ and recursively determine $E := \textsc{SmEll}(P \backslash \{q\})$. If $q \in E$, then $E = \textsc{SmEll}(P)$ and we are done. Otherwise, q must lie on the boundary of $\textsc{SmEll}(P)$, and we get $\textsc{SmEll}(P) = \textsc{SmEll}(P \setminus \{q\}, \{q\})$, the smallest enclosing ellipsoid of $P \setminus \{q\}$ with q on the boundary (Figure 1). Computing the latter (in the same way) is now an easier task because one degree of freedom has been eliminated. The generic call of the algorithm computes $\textsc{SmEll}(Q, R)$, the smallest ellipsoid enclosing Q that has R on the boundary. Before we give a detailed description, let us state a few important facts (proofs of which may be found in [16, 11]).

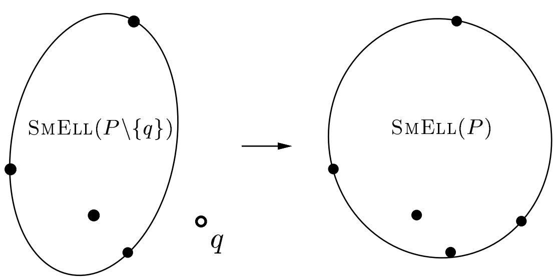

Figure 1: The inductive step in Welzl's algorithm

Proposition 2.1
(i) If there is any ellipsoid with R on its boundary that encloses Q, then $\textsc{SmEll}(Q, R)$ *exists and is unique.*
(ii) If $E =$ $\textsc{SmEll}(Q, R)$ *exists and $q \notin$* $\textsc{SmEll}(Q \setminus \{q\}, P)$*, then* $\textsc{SmEll}(Q \setminus \{q\}, R \cup \{q\})$ *exists and equals* $\textsc{SmEll}(Q, R)$.
(iii) If $\textsc{SmEll}(Q, R)$ *exists, then there is $S \subseteq Q$ with $|S| \leq \max(0, d(d+3)/2 - |R|)$ and* $\textsc{SmEll}(Q, R) = \textsc{SmEll}(S, R) = \textsc{SmEll}(\emptyset, S \cup R)$.

By (iii), a smallest enclosing ellipsoid is always determined by at most $\delta := d(d+3)/2$ *support points.* Incidentally, δ is the number of free variables in the ellipsoid parameters A and c.

If the point q to be removed for the recursive call is chosen *uniformly at random* among the points in Q, we arrive at the following randomized procedure.

Algorithm 2.2 *(computes* $\textsc{SmEll}(Q, R)$*, if it exists)*

```
SmEll(Q, R):
  IF Q = ∅ OR |R| = δ THEN
    RETURN SmEll(∅, R)
  ELSE
    choose q ∈ Q uniformly at random
    E := SmEll(Q \ {q}, R)
    IF q ∈ E THEN
      RETURN E
    ELSE
      RETURN SmEll(Q \ {q}, R ∪ {q})
    END
  END
```

To compute $\textsc{SmEll}(P)$, we call the algorithm with the pair $(P, \emptyset)$. Termination of the procedure is immediate because the recursive calls decrease the size of Q. Correctness follows from the proposition and the observation that the algorithm – when called with $(P, \emptyset)$ – maintains the invariant '$\textsc{SmEll}(Q, R)$ exists'. To justify the termination criterion '$|R| = \delta$', we need the following lemma proving that in this case only one ellipsoid E with R on the boundary exists, so that we must have $E = \textsc{SmEll}(\emptyset, R) = \textsc{SmEll}(Q, R)$. This is remarkable, because in general, an ellipsoid is *not* uniquely determined by any δ points on the boundary (for example, consider $\delta - 1$ points on the boundary of a $(d-1)$-dimensional ellipsoid E' and some additional point q; then there are many d-dimensional ellipsoids through E and q).

Lemma 2.3 *Whenever R attains cardinality δ during a call to* $\textsc{SmEll}(P, \emptyset)$*, exactly one ellipsoid E with R on its boundary exists.*

Proof. By expanding (1), we see that an ellipsoid is a special *second order surface* of the form

$$\{p \in \mathbb{R}^d \mid p^T M p + 2p^T m + w = 0\},$$

defined by $\delta + 1$ parameters $M \in \mathbb{R}^{d \times d}$ (symmetric), $m \in \mathbb{R}^d, w \in \mathbb{R}$.

For a point set $R \subseteq \mathbb{R}^d$ let $\mathcal{S}(R)$ denote the set of $(\delta+1)$-tuples of parameters that define second order surfaces through all points in R. It is clear that $\mathcal{S}(R)$ is a vector space, and we define the *degree of freedom* w.r.t. R to be $\dim(\mathcal{S}(R)) - 1$. Obviously, the degree of freedom is at least $\delta - |R|$, since any point in R introduces one linear relation between the parameters.

We now claim that during Algorithm 2.2, the degree of freedom w.r.t. R is always exactly $\delta - |R|$. This is clear for $R = \emptyset$. Moreover, if q is added to R in the second recursive call of the algorithm, the degree of freedom goes down, which proves the claim. To see this, assume on the contrary that $\dim(\mathcal{S}(R)) = \dim(\mathcal{S}(R \cup \{q\}))$, hence $\mathcal{S}(R) = \mathcal{S}(R \cup \{q\})$. Then it follows that q already lies on any second order surface through R, in particular on $\textsc{SmEll}(Q \setminus \{q\}, R)$. But then the second recursive call would not have been made, a contradiction.

Now the claim of the lemma follows: if $|R| = \delta$, the degree of freedom is 0, i.e. $\mathcal{S}(R)$ has dimension 1. Since a second order surface is invariant under scaling its parameters, this means that there is a unique second order surface, in this case an ellipsoid, through R. $\square$

To measure the expected performance of the algorithm, we count the number of *primitive* operations. These are the *ellipsoid computations* ('$\text{SMELL}(\emptyset, R)$') and the *in-ellipsoid tests* ('$q \in E$'). In the subsequent sections we concentrate on these primitive operations in the case $d = 2$. For the sake of this section, let us adopt the asymptotic point of view: if d is constant, the primitive operations can be implemented in constant time as well, so their overall number determines the actual runtime of Algorithm 2.2 up to a constant multiple.

Let $c_j(m)$ (resp. $t_j(m)$) denote the expected number of ellipsoid computations (resp. in-ellipsoid tests) in a call to $\text{SMELL}(Q, R)$ with $|Q| = m$ and $|R| = \delta - j$. We get $c_j(0) = c_0(m) = 1, t_j(0) = t_0(m) = 0$, and for $m, j > 0$

$$
\begin{aligned}
c_j(m) &\le c_j(m-1) + \frac{j}{m} c_{j-1}(m-1), \\
t_j(m) &\le t_j(m-1) + 1 + \frac{j}{m} t_{j-1}(m-1),
\end{aligned}
$$

where j/m is an upper bound for the probability of making the second recursive call. Why? Choose S according to Proposition 2.1 (iii). We have $|S| \le j$, and the second recursive call becomes necessary only if $q \in S$. By induction one can show that

$$c_j(m) \le (1 + H_m)^j \le (2 + \ln m)^j,$$

$H_m = 1 + 1/2 + \cdots + 1/m$ the m-th harmonic number, and

$$t_j(m) \le \left(\sum_{k=1}^{j} \frac{1}{k!} \right) j!m \le (e-1)j!m,$$

e the Euler constant. Thus, the expected number of primitive operations necessary to compute $\text{SMELL}(P)$ is bounded by

$$c_\delta(n) + t_\delta(n) \le (2 + \ln n)^\delta + (e-1)\delta! n,$$

which is $O(n)$ for constant d.

The move-to-front heuristic. There are point sets on which the algorithm does not perform substantially better than expected; on such point sets, the exponential behavior in $\delta = \Theta(d^2)$ leads to slow implementations already for small d. Although for $d = 2$ the actual runtime is still tolerable for moderately large n, a dramatic improvement (leading to a practically efficient solution for large n as well) is obtained under the so-called *move-to-front* heuristic. This variant keeps the points in an ordered list (initially random). In the first recursive call, q is chosen to be the last point in the list (restricted to the current subset of the points). If the subsequent in-ellipsoid test reveals $q \notin \text{SMELL}(Q \setminus \{q\}, R)$, p is moved to the front of the list, after the second recursive call to $\text{SMELL}(Q \setminus \{q\}, R \cup \{q\})$ has been completed. See [22] for further details and computing times.

3 Conics

In the sequel we elaborate on the primitive operations of Welzl's algorithm in the case $d = 2$. To prepare the ground, we look at ellipses from the more general perspective of arbitrary conics.

A *conic* $\mathcal{C}$ (second order curve, quadratic form) in *linear form* is the set of points $p = (x, y)^T \in \mathbb{R}^2$ satisfying the quadratic equation

$$\mathcal{C}(p) := rx^2 + sy^2 + 2txy + 2ux + 2vy + w = 0, \tag{2}$$

r, s, t, u, v, w being real parameters. Note that $\mathcal{C}$ is invariant under scaling the vector (r, s, t, u, v, w) by any nonzero factor. After setting

$$M := \begin{pmatrix} r & t \\ t & s \end{pmatrix}, m := \begin{pmatrix} u \\ v \end{pmatrix},$$

the conic assumes the form

$$\mathcal{C} = \{p^T M p + 2p^T m + w = 0\}. \tag{3}$$

If a point $c \in \mathbb{R}^2$ exists such that $Mc = -m$, $\mathcal{C}$ is symmetric about c and can be written in *center form* as

$$\mathcal{C} = \{(p - c)^T M (p - c) - z = 0\}, \tag{4}$$

where $z = c^T Mc - w$. If $\det(M) \neq 0$, a center exists and is unique. Conics with $\det(M) > 0$ are *ellipses*, for $\det(M) < 0$ we get *hyperbolas*. If $\det(M) = 0$, $\mathcal{C}$ defines a *parabola* which has a center (and then infinitely many) only in the case where $\mathcal{C}$ degenerates to a pair of parallel lines.

If $\mathcal{C}$ is an ellipse, i.e. $\det(M) > 0$, we can without loss of generality assume that M is positive definite (which in the $2d$-case just means $\det(M) > 0$ and $r, s > 0$). For this, note that $\det(M) > 0$ implies that r, s have the same sign, so (2) can be scaled in such a way that r, s both become positive. In this case, a point $q = (x, y)^T$ lies inside resp. outside the ellipse if $\mathcal{C}(q) \leq 0$ resp. $\mathcal{C}(q) > 0$. Also, if $\mathcal{C}$ is in center form (4), then we either have $z = 0$ in which case $\mathcal{C}$ is the trivial ellipse $\mathcal{C} = \{c\}$, or $z > 0$ holds. In the latter case, we may as well assume $z = 1$, after scaling M and z accordingly. This takes us back to the form of (1), with $A := M/z$.

Of particular importance is the *linear combination* of conics. If $\mathcal{C}_1, \mathcal{C}_2$ are two conics, the linear combination $\mathcal{C} := \lambda\mathcal{C}_1 + \mu\mathcal{C}_2$ is given by $\mathcal{C}(p) = \lambda\mathcal{C}_1(p) + \mu\mathcal{C}_2(p), \lambda, \mu \in \mathbb{R}$. Obviously, if a point q belongs to both $\mathcal{C}_1$ and $\mathcal{C}_2$, q also belongs to $\mathcal{C}$.

Conics through four points. A unique conic goes through any five points $p_1, \ldots, p_5$ (see e.g. [19]), while any four points $p_1, \ldots, p_4$ determine a one-dimensional bundle of conics, which is given as the linear span of two particular conics $\mathcal{C}_1, \mathcal{C}_2$. We may choose $\mathcal{C}_1$ as the pair of lines $\overline{p_1p_2}$ and $\overline{p_3p_4}$, and $\mathcal{C}_2$ as the pair of lines $\overline{p_2p_3}$ and $\overline{p_4p_1}$, see Figure 2.

These are indeed conics (namely degenerate hyperbolas) which can be seen as follows.

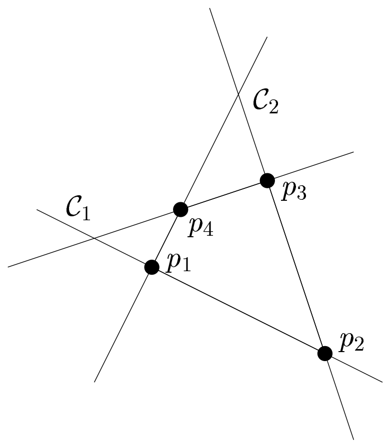

Figure 2: Two special conics through four points

For three points $q_1 = (x_1, y_1), q_2 = (x_2, y_2), q_3 = (x_3, y_3)$, define

$$[q_1q_2q_3] := \det \begin{pmatrix} x_1 - x_3 & x_2 - x_3 \\ y_1 - y_3 & y_2 - y_3 \end{pmatrix}.$$

It is well-known (and easy to verify) that $[q_1q_2q_3]$ records the orientation of the point triple. Let ℓ be the oriented line through q_1 and q_2. It holds that

$$q_3 \text{ lies} \left\{ \begin{array}{c} \text{to the left of} \\ \text{on} \\ \text{to the right of} \end{array} \right\} \ell \iff [q_1q_2q_3] \left\{ \begin{array}{l} > 0 \\ = 0 \\ < 0 \end{array} \right. .$$

Consequently,

$$\mathcal{C}_1(p) = [p_1p_2p][p_3p_4p], \quad \mathcal{C}_2(p) = [p_2p_3p][p_4p_1p],$$

and these turn out to be quadratic expressions as required in the conic equation (2).

Given another point q, the unique conic through p_1, p_2, p_3, p_4, q is easily computed as $\lambda\mathcal{C}_1 + \mu\mathcal{C}_2$, with

$$\lambda = \mathcal{C}_2(q), \quad \mu = -\mathcal{C}_1(q). \tag{5}$$

4 Primitive Operations

In the planar case, the primitive operations of Welzl's algorithm are

- computation of the smallest ellipse with k points on the boundary, $k \leq 5$, and
- in-ellipse tests involving ellipses of the former kind and arbitrary points.

Recall that for $k \leq 2$ the ellipse is degenerate, i.e. it is the empty set, a point or a segment, in which case the tests are easy. Subsequently we assume $3 \leq k \leq 5$.

How can we tell the area of an ellipse E? If E is given in center form (1), we have

$$\mathrm{Vol}(E) = \frac{\pi}{\sqrt{\det(A)}}, \tag{6}$$

which can easily be seen by choosing the coordinate system according to the principal axes of E, such that A becomes diagonal, see e.g. [16]. This means, in order to minimize the area, we have to maximize $\det(A)$.

We pursue different approaches, depending on the value of k. First recall that Welzl's algorithm guarantees that some ellipse through the given set of boundary points exists. In particular, the points are in convex position. For $k = 3$, the smallest ellipse has a rational representation and turns out to be easy to compute in center form, see below. If $k = 5$, the smallest ellipse is just the unique conic $\mathcal{C}$ through these points, and we have shown in Section 3 how to find the linear form of $\mathcal{C}$, see (5). In both cases we get explicit rational coordinates, so in-ellipse tests are straightforward.

The difficult case is $k = 4$. The smallest ellipse might have irrational coordinates, making exact in-ellipse tests nontrivial. One could try to work with a symbolic representation of the ellipse in terms of explicit algebraic numbers (such a representation always exists); however, the computation of this representation as well as the subsequent in-ellipse tests over it are difficult, computationally expensive, and – as we will see – unnecessary. Rather, we present a method to decide the in-ellipse test without knowing the ellipse; all we need to know are the four support points, implicitly representing the ellipse.

Before we start with this, let us do an example involving four concrete points. On the one hand, this illustrates the notions introduced so far, on the other hand it shows that the smallest ellipse may indeed have irrational coordinates.

Consider the points $p_1 = (0,0)^T$, $p_2 = (1,0)^T$, $p_3 = (1/2,1)^T$, and $p_4 = (0,1)^T$, see Figure 3 (we always assume that the four points are given in counterclockwise order, as already anticipated by Figure 2). The two conics $\mathcal{C}_1(p) = [p_1p_2p][p_3p_4p]$ and $\mathcal{C}_2(p) = [p_2p_3p][p_4p_1p]$ assume the form

$$\begin{aligned}\mathcal{C}_1(p) &= -y^2/2 + y/2,\\ \mathcal{C}_2(p) &= x^2 + xy/2 - x,\end{aligned}$$

and the linear combination $\mathcal{C} = \lambda\mathcal{C}_1 + \mu\mathcal{C}_2$ is obtained as

$$\mathcal{C}(p) = \mu x^2 - \frac{\lambda}{2}y^2 + \frac{\mu}{2}xy - \mu x + \frac{\lambda}{2}y.$$

Thus, in the form of (3), $\mathcal{C}$ is defined by

$$M = \begin{pmatrix} \mu & \mu/4 \\ \mu/4 & -\lambda/2 \end{pmatrix}, \quad m = \begin{pmatrix} -\mu/2 \\ \lambda/4 \end{pmatrix}, \quad w = 0.$$

If $\mathcal{C}$ is an ellipse, M is regular, and the center c is obtained as

$$c = M^{-1}m = \frac{1}{8\lambda + \mu}\begin{pmatrix} 3\lambda \\ 4\lambda + 2\mu \end{pmatrix}.$$

This leads to the center form (4), with

$$z = c^T M c = \frac{\lambda(\mu - \lambda)}{8\lambda + \mu}.$$

The area of the ellipse is minimized if

$$D(\lambda, \mu) := \det(A) = \det(M/z) = -\frac{1}{16}\frac{\mu(8\lambda+\mu)^3}{\lambda^2(\mu-\lambda)^2}$$

is maximized. Thus, the smallest ellipse is determined by values λ, μ such that the gradient ∇D vanishes. This happens if $8\lambda = -\mu$ (where we get $D = 0$) and if $4\lambda = (-3 \pm \sqrt{13})\mu$. In case of '+', $D \approx -40.93$ is obtained (and $\mathcal{C}$ is a hyperbola), in case of '−' we get $D \approx 5.93$, showing that any pair of nonzero coefficients (λ, μ) with $4\lambda = -(3+\sqrt{13})\mu$ determines the smallest ellipse with $p_1, \dots, p_4$ on the boundary. Note that no matter how λ, μ are scaled, the linear form of this ellipse contains irrational coefficients. The same is true for the center form. In particular, the center evaluates to $c = (x_c, y_c)$ with

$$x_c = \frac{9 + 3\sqrt{13}}{20 + 8\sqrt{13}} \approx .406, \quad y_c = \frac{1+\sqrt{13}}{5+2\sqrt{13}} \approx .377,$$

see Figure 3.

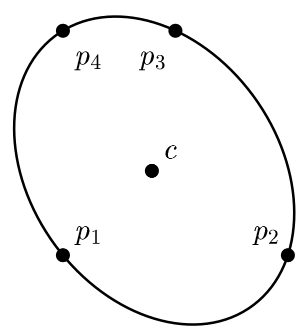

Figure 3: Irrational smallest ellipse through four points

4.1 Three Support Points

The smallest ellipse with three points on the boundary is represented in center form (4). We conceptually apply an affine transformation T on $S = \{p_1, p_2, p_3\}$, such that the triangle Δ with vertices $T(p_1), T(p_2), T(p_3)$ is equilateral. Since the affine transformation T scales any area by $\det(T)$, the smallest ellipse through S is transformed into the smallest ellipse through $T(S)$. This ellipse exists and is therefore unique (Proposition 2.1 (i)), hence it is the circumcircle of Δ. Applying the inverse transformation to the circumcircle yields

$$c = \frac{1}{3}\sum_{i=1}^{3} p_i, \quad M^{-1} = \frac{1}{3}\sum_{i=1}^{3}(p_i - c)(p_i - c)^T, \quad z = 2. \tag{7}$$

For details see [16]. The same formulae follow from design theory by consideration of the dual problem [20].

The in-ellipse test with point q is done by evaluating the sign of $\mathcal{C}(q) = (q-c)^T M(q-c)-z$. If and only if $\mathcal{C}(q) \leq 0$ then $q \in E$. For this, note that (7) always yields a positive definite matrix M.

4.2 Four Support Points

Ellipse computation. The smallest ellipse with four points on the boundary is not represented explicitly (remember that the coefficients of either representation may be irrational). Instead we represent the bundle of conics $\mathcal{C}$ through the four points $p_i = (x_i, y_i)^T, i = 1,2,3,4$ using $\mathcal{C}_1$ and $\mathcal{C}_2$ with

$$\mathcal{C}_1(p) := [p_1p_2p][p_3p_4p], \quad \mathcal{C}_2(p) := [p_2p_3p][p_4p_1p].$$

In the linear form (3), $\mathcal{C}_1$ and $\mathcal{C}_2$ have matrices

$$M_1 = \begin{pmatrix} r_1 & t_1 \\ t_1 & s_1 \end{pmatrix}, \quad M_2 = \begin{pmatrix} r_2 & t_2 \\ t_2 & s_2 \end{pmatrix}$$

with

$$\begin{aligned} r_1 &= (y_1 - y_2)(y_3 - y_4), \\ s_1 &= (x_1 - x_2)(x_3 - x_4), \\ t_1 &= -((x_1 - x_2)(y_3 - y_4) + (y_1 - y_2)(x_3 - x_4))/2 \end{aligned}$$

and

$$\begin{aligned} r_2 &= (y_2 - y_3)(y_4 - y_1), \\ s_2 &= (x_2 - x_3)(x_4 - x_1), \\ t_2 &= -((x_2 - x_3)(y_4 - y_1) + (y_2 - y_3)(x_4 - x_1))/2. \end{aligned}$$

Defining

$$\begin{aligned} \sigma_1 &:= ((x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4))/2, \\ \sigma_2 &:= ((x_2 - x_3)(y_4 - y_1) - (y_2 - y_3)(x_4 - x_1))/2 \end{aligned} \tag{8}$$

results in

$$\det(M_i) = r_i s_i - t_i^2 = -\sigma_i^2 \leq 0, \quad i = 1,2.$$

This shows, that the conics $\mathcal{C}_1$ and $\mathcal{C}_2$ are indeed hyperbolas (or pairs of parallel lines, i.e. degenerate parabolas).

As mentioned in Section 3, the type of $\mathcal{C} = \lambda\mathcal{C}_1 + \mu\mathcal{C}_2$ is determined by the sign of $\det(M)$, $M = \lambda M_1 + \mu M_2$, and we get

$$\begin{aligned} \det(M) &= \det(\lambda M_1 + \mu M_2) \\ &= (r_1s_1 - t_1^2)\lambda^2 + (r_2s_2 - t_2^2)\mu^2 + (r_1s_2 + r_2s_1 - 2t_1t_2)\lambda\mu \\ &= \alpha\lambda^2 + \gamma\mu^2 + \beta\lambda\mu, \end{aligned} \tag{9}$$

with $\alpha := \det(M_1)$, $\gamma := \det(M_2)$ and $\beta := r_1s_2 + r_2s_1 - 2t_1t_2$.

In-ellipse test. To test a query point q, we first compute the unique conic

$$\mathcal{C}_0 = \lambda_0 \mathcal{C}_1 + \mu_0 \mathcal{C}_2$$

through the five points p_1, p_2, p_3, p_4, q according to (5) and determine its type via $\det(M_0) = \det(\lambda_0 M_1 + \mu_0 M_2)$. We distinguish two cases, concerning the type of $\mathcal{C}_0$.

Case 1: Hyperbola/Parabola. This case is easy, as the following lemma shows.

Lemma 4.1 *If $\mathcal{C}_0$ is not an ellipse, then exactly one of the following holds.*
(i) q lies inside all ellipses through p_1, p_2, p_3, p_4.
(ii) q lies outside all ellipses through p_1, p_2, p_3, p_4.

Let us give some intuition, before we formally prove the lemma. Since no three of the four support points are collinear, there exist two (possibly degenerate) parabolas through these points (see Figure 4). These parabolas cut the plane into regions which determine the type of $\mathcal{C}_0$. Only if q lies strictly inside one parabola and strictly outside the other, $\mathcal{C}_0$ is an ellipse. Otherwise, q either lies inside both parabolas in which case q also lies inside all ellipses through p_1, p_2, p_3, p_4, or q lies outside both parabolas, also being outside all the ellipses.

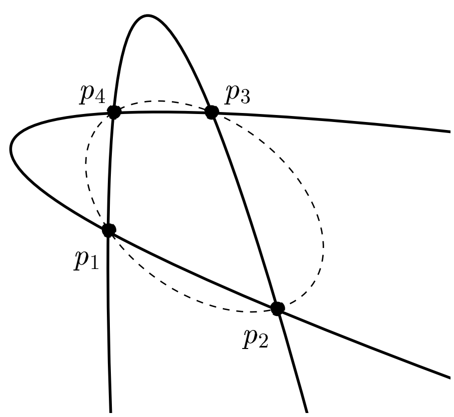

Figure 4: The two parabolas through four points

Proof. Assume there exist two ellipses E_{in} and E_{out} through the four points, E_{in} containing q, E_{out} not containing q, equivalently $E_{in}(q) \leq 0$, $E_{out}(q) > 0$. Then choose $\lambda \in [0, 1)$ such that

$$E(q) := (1 - \lambda) E_{in}(q) + \lambda E_{out}(q) = 0.$$

Since the convex combination of two positive definite matrices is also positive definite, E is an ellipse through p_1, p_2, p_3, p_4, q; thus $E = \mathcal{C}_0$ holds, which is a contradiction to $\mathcal{C}_0$ being not an ellipse. $\square$

Consequently, p either lies inside all ellipses or outside all ellipses through the four support points. Thus, all we need to do is test q against an arbitrary ellipse through the four support points. To get such an ellipse, we choose the linear combination $\lambda\mathcal{C}_1 + \mu\mathcal{C}_2$ with coefficients

$$\lambda := 2\gamma - \beta, \quad \mu := 2\alpha - \beta,$$

which by (9) gives

$$\det(M) = (4\alpha\gamma - \beta^2)(\alpha + \gamma - \beta)$$

We will show that both factors have negative sign, thus proving that the choice of λ and μ indeed yields an ellipse E. To see this, we first check that

$$4\alpha\gamma - \beta^2 = -[p_1p_2p_3][p_2p_3p_4][p_3p_4p_1][p_4p_1p_2].$$

Since p_1, p_2, p_3, p_4 are in counterclockwise order, each bracketed term has positive sign, i.e. $4\alpha\gamma - \beta^2 < 0$ holds. On the other hand, we have

$$\alpha + \gamma - \beta = [p_2p_4p_1][p_2p_4p_3] - (\sigma_1 + \sigma_2)^2,$$

with σ_1, σ_2 as defined in (8). The first term is negative because p_1 and p_3 lie on different sides of the diagonal $\overline{p_2p_4}$. It follows that $\alpha + \gamma - \beta < 0$ and finally $\det(M) > 0$.

If M is not yet positive definite, we scale E by -1. Then q lies inside E (and hence inside the smallest ellipse through p_1, p_2, p_3, p_4) if and only if $E(q) \leq 0$.

Case 2: Ellipse. $\mathcal{C}_0$ is an ellipse E, and we need to check the position of q relative to E^*, the smallest ellipse through p_1, p_2, p_3, p_4, given as

$$E^* = \lambda^*\mathcal{C}_1 + \mu^*\mathcal{C}_2,$$

with unknown parameters λ^*, μ^*. In the form of (2), E is given by $(r_0, s_0, t_0, u_0, v_0, w_0)$, where

$$r_0 = \lambda_0 r_1 + \mu_0 r_2,$$

r_1 and r_2 the respective parameters of $\mathcal{C}_1$ and $\mathcal{C}_2$. By scaling the representation of E^* accordingly, we can also assume that

$$r_0 = \lambda^* r_1 + \mu^* r_2$$

holds. In other words, E^* is obtained from E by varying its parameters λ_0, μ_0 along the line $\{\lambda r_1 + \mu r_2 = r_0\}$,

$$\begin{pmatrix} \lambda^* \\ \mu^* \end{pmatrix} = \begin{pmatrix} \lambda_0 \\ \mu_0 \end{pmatrix} + \tau^* \begin{pmatrix} -r_2 \\ r_1 \end{pmatrix}. \tag{10}$$

Define

$$E^\tau := (\lambda_0 - \tau r_2)\mathcal{C}_1 + (\mu_0 + \tau r_1)\mathcal{C}_2.$$

Then $E^0 = E, E^{\tau^*} = E^*$. The function $g(\tau) = E^\tau(q)$ is linear, hence we get

$$E^*(q) = \tau^* \left.\frac{\partial}{\partial\tau} E^\tau(q)\right|_{\tau=0} = \rho\,\tau^*,$$

$\rho = \mathcal{C}_2(q)r_1 - \mathcal{C}_1(q)r_2$. Assuming that E is scaled such that $r_0 > 0$, this means that q lies inside E^* iff $\rho\tau^* \leq 0$.

It remains to determine the sign of τ^*, in other words: starting from E, 'in which direction' lies E^*? The following lemma has been proved in [5], see also [16].

Lemma 4.2 *Consider two ellipses E_1, E_2, and let*

$$E^\lambda = (1-\lambda)E_1 + \lambda E_2$$

be their convex combination, $\lambda \in [0,1]$. Then E^λ is an ellipse satisfying

$$\mathrm{Vol}(E^\lambda) < \max(\mathrm{Vol}(E_1), \mathrm{Vol}(E_2)),$$

for all $\lambda \in (0,1)$.

Since E^τ is a convex combination of E and E^* for τ ranging between 0 and τ^*, the volume of E^τ decreases as τ goes from 0 to τ^*, hence

$$\mathrm{sgn}(\tau^*) = -\mathrm{sgn}\left(\left.\frac{\partial}{\partial\tau}\mathrm{Vol}(E^\tau)\right|_{\tau=0}\right).$$

If A^τ is the matrix of E^τ in center form (1), the volume formula (6) gives

$$\mathrm{sgn}\left(\left.\frac{\partial}{\partial\tau}\mathrm{Vol}(E^\tau)\right|_{\tau=0}\right) = -\mathrm{sgn}\left(\left.\frac{\partial}{\partial\tau}\det(A^\tau)\right|_{\tau=0}\right),$$

and $\det(A^\tau)$ is easily expressed as a function of τ. For this, recall that if M, m, w are the parameters of E^τ in the form of (3), $c = M^{-1}m$ the center, we get

$$A^\tau = M/z, \quad z = m^T M^{-1} m - w,$$

where M, m, w are functions of τ (which we omit in the sequel for the sake of readability). Noting that

$$M^{-1} = \frac{1}{\det(M)} \begin{pmatrix} s & -t \\ -t & r \end{pmatrix},$$

we get

$$z = \frac{1}{\det(M)}(u^2 s - 2uvt + v^2 r) - w.$$

Let us introduce the following abbreviations.

$$d := \det(M), \quad Z := u^2 s - 2uvt + v^2 r.$$

With primes (d', Z' etc.) we denote derivatives w.r.t. τ. Now we can write the derivative in question as

$$\frac{\partial}{\partial\tau}\det(A^{\tau}) = \frac{\partial}{\partial\tau}\det(M/z) = (d/z^2)' = \frac{d'z - 2dz'}{z^3}. \qquad (11)$$

Since $d(0), z(0) > 0$ (recall that these values refer to the ellipse $E^0 = E$), this is equal in sign to

$$\delta := d(d'z - 2dz'),$$

at least when evaluated for $\tau = 0$, which is the value we are interested in. Furthermore, we have

$$\begin{aligned} d'z &= d'(\frac{1}{d}Z - w) = \frac{d'}{d}Z - d'w, \\ dz' &= d(\frac{Z'd - Zd'}{d^2} - w') = \frac{Z'd - Zd'}{d} - dw'. \end{aligned}$$

Hence

$$\begin{aligned} \delta &= d'Z - dd'w - 2(Z'd - Zd' - d^2w') \\ &= 3d'Z + d(2dw' - d'w - 2Z'). \end{aligned}$$

Rewriting Z as $u(us - vt) + v(vr - ut) = uZ_1 + vZ_2$, we get

$$\begin{aligned} d &= rs - t^2, & Z_1' &= u's + us' - v't - vt', \\ d' &= r's + rs' - 2tt', & Z_2' &= v'r + vr' - u't - ut', \end{aligned}$$

and finally

$$Z' = u'Z_1 + uZ_1' + v'Z_2 + vZ_2'.$$

For $\tau = 0$, all these values can be computed directly from $r(0), \ldots, w(0)$ (the defining values of E) and their corresponding primed values $r'(0), \ldots w'(0)$. For the latter we get $r'(0) = 0, s'(0) = r_1s_2 - r_2s_1, \ldots, w'(0) = r_1w_2 - r_2w_1$ ($r_i, \ldots, w_i$ the defining values of $\mathcal{C}_i, i = 1, 2$). Summarizing, we obtain that q lies inside E^* iff $\operatorname{sgn}(\rho\ \delta(0)) \leq 0$.

4.3 Five Support Points

It is not difficult to check that in Welzl's algorithm 2.2, R attains cardinality five only if immediately before, a test '$p \in E$' has been performed (with a negative result), where E is determined by four support points. In the process of doing this test, the unique conic (which we know is an ellipse E) through the five points has already been computed, see previous section. Thus, given another point q, we can just 'recycle' E for the in-ellipse test with q.

5 Implementation

We have implemented the in-ellipse tests as subroutines of Welzl's method with move-to-front heuristic [22], without any tuning so far.[3] On a Sun SPARC-station 20, using rational arithmetic over LEDA's arbitrary length integers, the algorithm takes 220 seconds to compute $\mathrm{SMELL}(P)$, P a set of 10,000 points with random 32-bit integer coordinates. Under floating-point arithmetic, the computing time drops to 2 seconds, but the result might be incorrect. This gap (suggesting successful usage of floating-point filters and other techniques to combine fast arithmetic with exact computation) is explained by the fact that numbers get large under rational arithmetic. If the input coordinates are b-bit integers, an exact evaluation of $\delta(0)$ as in the previous section requires $30b + O(1)$ bits of precision in the worst case.

The algorithm's output is a support set S, $3 \leq |S| \leq 5$, such that $\mathrm{SMELL}(P) = \mathrm{SMELL}(S)$. In addition, for $|S| \neq 4$, our method determines $\mathrm{SMELL}(S)$ explicitly. For $|S| = 4$, the value of τ^* defining $\mathrm{SMELL}(S)$ via (10) appears among the roots of (11); a careful analysis [14, 16] reduces this to a cubic polynomial in τ, thus an exact symbolic representation or a floating-point approximation of τ^* and $\mathrm{SMELL}(S)$ can be computed in a postprocessing step.

6 Discussion

We have described an exact $O(n)$ algorithm for computing the smallest enclosing ellipse of a planar point set, obtained by implementing the primitives of Welzl's method in rational arithmetic.

From a practical point of view, the three-dimensional version of the problem is probably most interesting, and one might ask how our techniques apply to this case. While Welzl's algorithm as described in Section 2 works in any dimension, the primitive operations are already not sufficiently understood for $d = 3$. First of all, the number of basic cases is larger; we need to do in-ellipsoid tests over ellipsoids defined by $4 \leq k \leq 9$ boundary points. While the extreme cases $k = 4, 9$ are easy (they behave similar to the extreme cases $k = 3, 5$ for $d = 2$), no exact method for any other case is known. Our ideas readily generalize to the case $k = 8$: here we can (as in the planar case) use the fact that eight points – if they appear as a set R during Algorithm 2.2 – determine an ellipsoid up to one degree of freedom, see the proof of Lemma 2.3. Beyond that, it is not clear whether the method generalizes. In our fastest existing implementation for the case $d = 3$, we use the formerly mentioned gradient descent method of [21] to compute the required ellipsoids according to some prespecified accuracy, for $k = 5, \dots, 8$, see [16].

In any dimension larger than two, an open problem is to prove the existence of a *rational* expression whose sign tells whether a point $q \in \mathbb{R}^d$ lies inside the smallest ellipsoid determined by $d + 1 \leq k \leq d(d + 3)/2$ boundary points. If such an expression exists, how can it be computed, and what is its complexity?

[3] A tuned version will become part of the CGAL library, see `http://www.cs.ruu.nl/CGAL/`

Acknowledgments

We thank the *SoCG '97* referees for pointing out two errors in a previous version and for general comments that helped to improve the presentation. The figures were drawn using the interactive incidence geometry tool *Cinderella's Café* (`http://www.cinderella.de/`) by Jürgen Richter-Gebert and Ulrich Kortenkamp.

References

[1] I. Adler and R. Shamir. A randomization scheme for speeding up algorithms for linear and convex quadratic programming problems with a high constraints-to-variables ratio. *Math. Programming*, 61:39–52, 1993.

[2] V. Barnett. The ordering of multivariate data. *J. Roy. Statist. Soc.*, 139:318–354, 1976.

[3] C. Bouville. Bounding ellipsoids for ray-fractal intersection. In *SIGGRAPH*, pages 45–52, 1985.

[4] B. Chazelle. Approximation and decomposition of shapes. In J. T. Schwartz and C.-K. Yap, editors, *Advances in Robotics 1: Algorithmic and Geometric Aspects of Robotics*, pages 145–185. Lawrence Erlbaum Associates, Hillsdale, NJ, 1987.

[5] L. Danzer, D. Laugwitz, and H. Lenz. Über das Löwnersche Ellipsoid und sein Analogon unter den einem Eikörper eingeschriebenen Ellipsoiden. *Arch. Math.*, 8:214–219, 1957.

[6] M. E. Dyer. A class of convex programs with applications to computational geometry. In *Proc. 8th Annu. ACM Symp. on Computational Geometry*, pages 9–15, 1992.

[7] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice.* Addison-Wesley, Reading, MA, 1990.

[8] B. Gärtner and Sven Schönherr. Exact primitives for smallest enclosing ellipses. In *Proc. 13th Annu. ACM Symp. on Computational Geometry*, pages 430–432, 1997.

[9] B. Gärtner and E. Welzl. Linear programming – randomization and abstract frameworks. In *Proc. 13th Annu. Symp. on Theoretical Aspects of Computer Science (STACS)*, volume 1046 of *Lecture Notes in Computer Science*, pages 669–687. Springer-Verlag, 1996.

[10] F. John. Extremum problems with inequalities as subsidiary conditions. In *Studies and Essays presented to R. Courant on his 60th Birthday*, pages 187–204. Interscience Publishers, NY, 1948.

[11] K. Leichtweiß. Über die affine Exzentrizität konvexer Körper. *Arch. Math.*, 10:187–199, 1959.

[12] J. Matoušek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 1–8, 1992.

[13] K. Mehlhorn and S. Näher. LEDA: a platform for combinatorial and geometric computing. *Communications of the ACM*, 38:96–102, 1995.

[14] M. J. Post. Computing minimum spanning ellipses. Technical Report CS-82-16, Department of Computer Science, Brown University, 1982.

[15] M. J. Post. Minimum spanning ellipsoids. In *Proc. 16th Annu. ACM Sympos. Theory Comput.*, pages 108–116, 1984.

[16] S. Schönherr. Berechnung kleinster Ellipsoide um Punktemengen. Diploma thesis, Free University Berlin, 1994.

[17] N. Z. Shor and O. A. Berezovski. New algorithms for constructing optimal circumscribed and inscribed ellipsoids. *Opt. Methods and Software*, 1:283–299, 1992.

[18] B. W. Silverman and D. M. Titterington. Minimum covering ellipses. *SIAM J. Sci. Statist. Comput.*, 1:401–409, 1980.

[19] B. Spain. *Analytical Conics.* Pergamon Press, 1957.

[20] D. M. Titterington. Optimal design: some geometrical aspects of D-optimality. *Biometrika*, 62(2):313–320, 1975.

[21] D. M. Titterington. Estimation of correlation coefficients by ellipsoidal trimming. *Appl. Statist.*, 27(3):227–234, 1978.

[22] E. Welzl. Smallest enclosing disks (balls and ellipsoids). In H. Maurer, editor, *New Results and New Trends in Computer Science*, volume 555 of *Lecture Notes in Computer Science*, pages 359–370. Springer-Verlag, 1991.