

Checking the Convexity of Polytopes and the Planarity of Subdivisions* (Extended Abstract)

O. Devillers¹ G. Liotta² F. P. Preparata³ R. Tamassia³

¹ INRIA, BP 93, 06902 Sophia Antipolis Cedex, France
`odevil@sophia.inria.fr`

² Dipartimento di Informatica e Sistemistica, Università di Roma “La Sapienza”
via Salaria 113, 00198 Rome, Italy
`liotta@dis.uniroma1.it`

³ Center for Geometric Computing, Department of Computer Science
Brown University, Providence, RI 02912–1910, USA
`franco@cs.brown.edu` `rt@cs.brown.edu`

Abstract. This paper studies the problem of verifying the correctness of geometric structures. We design optimal checkers for convex polytopes in two and higher dimensions, and for various types of planar subdivisions, such as triangulations, Delaunay triangulations, and convex subdivisions. Our checkers are simpler and more general than the ones previously described in the literature. Their performance is studied also in terms of the degree, which characterizes the arithmetic precision required.

1 Introduction

The development of checkers for geometric structures is justified by the expectation that it is easier to evaluate the quality of the output than the correctness of the algorithm producing it, and is further motivated by the increasing availability of geometric software on the Internet (see, e.g., [1]), and by the emerging client-server distributed models of geometric computing over the Web (see, e.g., [2]). Mehlhorn *et al.* [19] identify three fundamental features of a good checker: *correctness*, *simplicity*, and *efficiency*.

In this paper, we consider checkers for subdivisions in two and higher dimensions. In particular, we consider two-dimensional planar subdivisions and convex polytopes in a fixed dimension d . The subdivision to be checked can be either a primitive structure, or a derived structure computed from a primitive one (e.g., the Voronoi diagram of a set of sites in the plane). More formally, we consider checkers whose input consists of:

- a geometric graph F (i.e., a graph with coordinates assigned to its vertices), which is claimed to induce a subdivision S ;
- an optional primitive structure P , from which S is claimed to be derived;

* Research supported in part by the U.S. Army Research Office under grant DAAH04–96–1–0013, by the National Science Foundation under grant CCR–9423847, and by the EC ESPRIT Long Term Research Project ALCOM-IT under contract 20244.

- an optional *certificate* C , provided to facilitate the task of the checker; and
- a predicate \mathcal{P} stating a property of S .

The task of the checker is to verify that a subdivision S induced by geometric graph Γ and satisfying predicate \mathcal{P} does indeed exist. The checker either *accepts* Γ , or *rejects* Γ producing evidence that no such subdivision S exists.

Consider the following two examples:

- Γ is a graph with (x, y) coordinates assigned to the vertices; P is a set of points in the plane; C is a circular ordering of the edges incident on each vertex of Γ ; and predicate \mathcal{P} states that S is the Voronoi diagram of P .
- Γ is a graph with (x, y, z) coordinates assigned to the vertices; P and C are not defined; and predicate \mathcal{P} states that S is a convex 3D polytope.

Clearly, the availability of a suitable certificate may be crucial to the efficiency of the checker. We consider three scenarios for the type of checker available:

Arbitrary-Certificate Scenario: This is the most favorable scenario for the checker. Namely, the checker can specify an arbitrary certificate C to be provided as additional input. The size $|C|$ of the certificate should be $O(|S|)$ if S is itself primitive, and $O(|S| + |P|)$ if S is derived from a primitive structure P . Also, if S is derived from P , the certificate should be computable as a byproduct of an optimal algorithm that constructs S from P without using additional asymptotic time or space. A checker that operates within the Arbitrary-Certificate Scenario is called an *A-checker*.

Consider, for example, the problem of verifying that a polygon Γ is the convex hull S of a set P of primitive points in the plane. A useful certificate C would consist, for each point p of P , either of a vertex of Γ coincident with p , or of a triplet a, b, c of vertices of Γ such that p is contained in the triangle $\Delta(a, b, c)$. It has been shown that there exists an A-checker using certificate C for verifying that a polygon Γ is the convex hull S of the points of a set P in linear time $O(|G| + |P|)$ (see, e.g., [21]).

Topology-Certificate Scenario: In this intermediate scenario, the checker has available a certificate C that describes the (claimed) topology of the subdivision S . A checker that operates within the Topology-Certificate Scenario is called a *T-checker*.

In the previous example of the planar convex-hull verification, a certificate C for the Topology-Certificate Scenario would consist simply of the circular sequence of the vertices of polygon Γ . With this certificate, it is possible to perform in linear time only a partial verification that polygon Γ is the convex hull S of the points of P . Namely, one can verify in time $O(|G|)$ that Γ is a convex polygon (see Section 3). However, verifying that the points of P that are not vertices of Γ are interior to Γ requires time $\Omega(\log |G|)$ per point.

For the more complex problem of verifying that a 3D geometric graph Γ realizes a convex polytope S , a certificate C for the Topology-Certificate Scenario would be a data structure that describes a planar embedding of Γ ,

e.g., circularly-sorted adjacency lists.

No-Certificate Scenario: This is the least favorable scenario for the checker. Namely, no certificate C is available to the checker, which must perform the verification using only geometric graph G . A checker that operates within the No-Certificate Scenario is called an *N-checker*.

The Arbitrary-Certificate Scenario follows the program checking paradigm pioneered by Blum and Kannan [4]. On the negative side, it requires that the algorithms constructing the subdivision be modified to produce the specified certificate. On the positive side, A-checkers are often faster and simpler to implement than other types of checkers, and their correctness is usually easily established. Sullivan *et al.* [21] show A-checkers for planar convex hull, sorting, and shortest path algorithms. A-checkers for d -dimensional convex hulls are also discussed in [19].

The Topology-Certificate Scenario requires a natural type of certificate, which many algorithms for constructing subdivisions are likely to produce by default. Mehlhorn *et al.* [19] present a T-checker for certifying the convexity of a d -dimensional polytope and mention T-checkers for several other geometric structures, including Delaunay triangulations and Voronoi diagrams.

The No-Certificate Scenario is likely to occur in various application contexts, such as CAD models, where 3D subdivisions are represented as “polygon soups”, i.e., collections polygons with no topological information [11]). Also, the availability of N-checkers is important when one tries to incorporate in a program modules developed by others, whose source code may be difficult to understand or modify.

This paper contains a systematic study of checkers for various types of planar subdivisions and for convex polytopes in two and higher dimensions. We advocate a new requirement that good geometric checkers should satisfy, and present simple and efficient N-checkers for several structures for which only A-checkers and T-checkers have been so far designed. Specifically, our contributions can be summarized as follows.

1. As an additional measure of effectiveness for a checker, we adopt the notion of *degree* [5, 16, 17], which takes into account the number of bits required by the checker to carry out error-free computations. A good checker should have degree no higher than the problem at hand allows. We give lower bounds on the degree of checkers for planar subdivisions and convex polytopes, and present optimal-degree checkers.
2. We present a new T-checker for convex polytopes that is simpler than the one given in [19]. Our T-checker works in any dimension and recursively reduces the verification of a d -dimensional polytope to the verification of an associated $(d - 1)$ -dimensional polytope. It is optimal with respect to both the time complexity and the degree. The design of our T-checker for convex polytopes reveals new combinatorial and geometric properties that may be of independent interest.
3. We present linear-time optimal degree T-checkers for triangulations, convex

subdivisions, and general planar subdivisions. Such checkers use as subroutines elementary graph algorithms and do not require to test the planarity of the underlying input graph.

4. Extending the above results on T-checkers, we present linear-time optimal degree N-checkers for triangulations and convex subdivisions. This solves significant special cases of an open problem mentioned by Kirkpatrick [14] on the existence of an $o(n \log n)$ algorithm to verify the planarity of a geometric graph.
5. As a further application, we give linear-time optimal-degree N-checkers for Delaunay triangulations, locally minimum-weight triangulations, and Delaunay diagrams. Finally, we give a linear-time optimal-degree N-checker for three-dimensional convex polytopes.

Near the completion of our investigations, we became aware of two ongoing projects on the design of T-checkers for planar subdivisions, including triangulations and convex subdivisions. A manual [18] describing the functionality of C++ functions that implement T-checkers for Delaunay triangulations, Voronoi diagrams and convex planar subdivisions is available from Mehlhorn’s Web page. A manuscript in progress [20] contains characterizations of triangulations and convex planar subdivisions similar to those of the present paper. There is therefore an undeniable shared objective between our research and that of Mehlhorn *et al.* Any minor overlaps of independently obtained results are largely offset by the fact that our proof techniques are different from theirs, and our checkers work also within the No-Certificate Scenario, while maintaining the same efficiency as the T-checkers of [18, 20].

The rest of this paper is organized as follows. Preliminaries and lower bounds on the degree of checkers are in Section 2. Our T-checker for convex polytopes is presented in Section 3. Section 4 is devoted to T-checkers for triangulations, convex subdivisions, and general planar subdivisions. N-checkers are studied in Section 5. Finally, open problems are given in Section 6. Due to space limitations, detailed proofs have been omitted in this extended abstract.

2 Preliminaries

We start with definitions for geometric graphs, ordered graphs, and planarity. We then recall the notion of degree of geometric algorithms, introduced in [16, 17] (a related concept is defined in [5]). Finally, we present lower bounds on the degree of checkers for convex polytopes and planar subdivisions.

A d -dimensional *geometric graph* is a graph drawn with straight-line edges in d -dimensional space, i.e., a graph whose vertices have d -dimensional coordinates. In the following, we often denote with Γ a geometric graph, and with G its underlying combinatorial structure. To simplify the notation and when the context is not misleading, we may denote with G both the geometric graph and its underlying combinatorial structure.

A two-dimensional geometric graph Γ is *planar* if it has no crossing edges, i.e., any two edges of Γ intersect only at a common vertex. For every planar

graph G , there exists a planar geometric graph Γ with underlying graph G , i.e., every planar graph admits a planar straight-line drawing (see, e.g., [10]). However, a geometric graph with an underlying planar graph is not necessarily planar.

A planar geometric graph Γ determines a *planar subdivision* S , i.e., a partition of the plane into regions called faces. Planar subdivision S is said to be *induced* by Γ . A planar subdivision is *convex* if the boundary of each face is a convex polygon. A planar subdivision S induced by geometric graph Γ is *maximal* if there is no other planar geometric graph Γ' such that Γ is a subgraph of Γ' . In a maximal planar subdivision, the boundary of each internal face is a triangle, and the boundary of the external face is a convex polygon. A maximal planar subdivision is also called a *triangulation*.

A graph G is *ordered* (or G has an *ordering*) if for each vertex v of G , a circular ordering of the edges incident on v is given. The ordering of a graph is usually denoted with Ψ . A two-dimensional geometric graph Γ has a natural ordering associated with it, called *the ordering of Γ* , given by the clockwise circular sequence of the edges incident on each vertex. An ordering Ψ of a graph induces a set of directed *circuits*, where the edge (v, w) following (u, v) in a circuit is the successor of (u, v) in the circular ordering of the edges incident on v . Every edge of the graph is traversed by exactly two circuits, once in each direction.

For a geometric graph Γ and its ordering Ψ , we call *outer circuit* the circuit induced by Ψ that contains the first edge following clockwise a horizontal rightward ray emanating from the topmost vertex of Γ , and *internal circuits* the remaining circuits induced by Ψ . If Γ is planar, then the circuits induced by Ψ are the boundaries of the faces of the subdivision S induced by Γ . In particular, the outer circuit is the boundary of the external face traversed clockwise, and the internal circuits are the boundaries of the internal faces traversed counterclockwise. The ordering Ψ of an ordered graph G is *planar* if there exists a planar geometric graph Γ whose underlying graph is G and whose ordering is Ψ . A planar ordering of G is associated with a planar (topological) embedding of G . A graph G admits a planar ordering only if it is planar. Also, the number of distinct planar orderings (topological embeddings) of a planar graph can be super-exponential.

An algorithm has *degree d* if its test computations involve the evaluation of multivariate polynomials of arithmetic degree at most d [16, 17]. We state two lower bounds on the degree of the checkers that will be studied in the rest of the paper. The proof is omitted in this extended abstract.

Theorem 1. *A checker for d -dimensional convex polytopes has degree at least d .*

Theorem 2. *A checker for planar subdivisions has degree at least 2.*

3 T-checkers for Convex Polytopes

In this section, we describe the design of a T-checker for convex polytopes in a fixed dimension d . The input to the checker is a geometric graph Γ in E^d , and

a certificate describing the topology of the polytope supposedly induced by Γ .

Because of its appeal to intuition, we consider the question for $d \leq 3$ (specifically for $d = 3$) and indicate later how to extend the result to higher dimensions. A 3-dimensional polytope has the topology of a planar graph and, without loss of generality, we may assume that all faces are triangles (simplicial polytope). In addition, for each edge a condition of convexity (local convexity) must be satisfied. Specifically for edge p_1p_2 ($p_i = (x_i, y_i, z_i)$), with incident facets $p_1p_2p_3$ and $p_1p_2p_4$, the simplex $(p_1p_2p_3p_4)$ (see Figure 1(a)) lies in the interior half-space of each of the two hyperplanes supporting the facets. This is equivalent to specifying that the sign of the determinant Δ_{1234} shown in Figure 1(b) is positive.

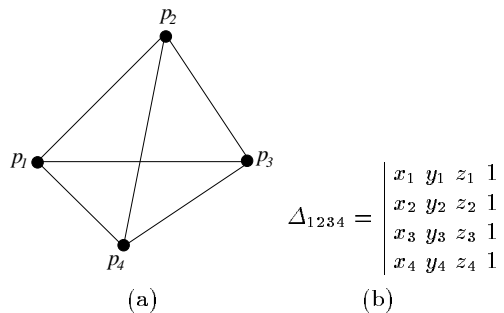


Fig. 1. (a) The simplex $(p_1 p_2 p_3 p_4)$. (b) Determinant Δ_{1234} .

It has been shown by Mehlhorn *et al* [19] that a locally convex plane-bounded surface (without boundary) is the surface of a convex polytope if and only if any ray from a point q in E^d lying on the negative side of all facets intersects the interior of exactly one facet. Therefore if a ray intersects the interiors of more than one facet, then the surface is not a convex polytope.

We now discuss an equivalent but simpler criterion. We will prove the validity of our criterion by showing its equivalence to the criterion of Mehlhorn *et al*. We say that an edge e of a 3-dimensional polytope Γ is a *seam edge* if a vertical (i.e., parallel to the z -axis) plane by e leaves both facets incident on e in the same closed half space and the interior of the upper facet in the open half space.

Lemma 3. *If polytope Γ is locally convex at each edge, then the subgraph Γ' of Γ induced by the seam edges of Γ is a collection of cycles.*

Assume now that there is a point q , on the interior side of each facet, such that a ray r from q intersects the interior of more that one facet, say f_1, f_2, \dots, f_s . Without loss of generality we discuss $s = 2$. Consider the vertical half-plane α containing r and the intersection of α with the surface defined by Γ , and let p_i be the intersection of r with f_i . Points p_1 and p_2 belong each to a closed curve in α (since Γ is without boundary). Starting from p_i and proceeding on

this curve by increasing z we shall traverse a polygonal curve. Since this curve is bounded, it will have a point of maximum z in the half-plane α , denoted u_i . Analogously, if we proceed from p_i by decreasing z we will reach a point of minimum z in α , denoted b_i . If u_i does not belong to the vertical line by q , then it admits a horizontal supporting in α ; otherwise the angle formed by the ascending polygonal line at u_i with the vertical by q is ≤ 0 . Analogously, b_i either admits a horizontal supporting plane or the angle of the polygonal line is ≥ 0 . We conclude that traversing the subchain $u_i b_i$ downwards we reach a point s_i which is the first to admit a vertical supporting plane, i.e., s_i belongs to the seam.

If we now project the seam and point q on the horizontal plane, point q projects to q' , half-plane α projects to a ray by q' , and each s_i projects to a point on this ray. Since intersection is preserved by projection, we conclude that any ray from q' intersects the projected seam σ' as many times as a ray from q intersects the surface of Γ .

We now show that for a locally convex polytope, the criterion of [19] translates into an analogous criterion applied to the projected seams, i.e., the projected seam edges must form a single cycle. The following lemma enables us to reduce by one the dimensionality of the criterion.

Lemma 4. *Let Γ be a locally convex polytope and let σ' be the collection of polygons in the xy -plane obtained by projecting the seam edges of Γ along the z -axis. Each cycle of σ' is locally convex.*

Sketch of Proof. Let $|w_1, w_2, w_3, w_4|$ denote the determinant whose i -th row is $[x(w_i), y(w_i), z(w_i), 1]$; $|w_1, w_2, w_3|$ is analogously defined. Let (u_1, v) and (v, u_s) be two seam edges incident on v and obviously consecutive in their cycle. If these two edges are also consecutive around v , then facet (u_1, v, u_2) trivially projects to a (locally convex) triangle. So for $s \geq 3$ let $(u_2, v), \dots, (u_{s-1}, v)$ be the intermediate edges clockwise around v (clockwise for an external observer). For $s=3$, by the local convexity test applied to edge (u_1, u_3) we conclude that segment (u_1, u_3) is internal to the polytope. For $s > 3$ the local convexity at (u_3, v) yields $|u_2, v, u_3, u_1| \geq 0$ and the local convexity at (u_4, v) yields $|u_3, v, u_4, u_2| \geq 0$. The latter is equivalent to $|u_2, v, u_3, u_4| \geq 0$. Now, for any real $0 \leq \alpha \leq 1$, we have:

$$\alpha |u_2, v, u_3, u_1| + (1 - \alpha) |u_2, v, u_3, u_4| \geq 0$$

i.e., $|u_2, v, u_3, \alpha u_1 + (1 - \alpha) u_4| \geq 0$, which shows that segment (u_1, u_4) is internal to the polytope (since any of its points is a convex combination of u_1 and u_4). Iterating the argument on u_5, \dots, u_s , we conclude that segment (u_1, u_s) is internal to the polytope, i.e., $|u_1, v, u_s, u_j| \geq 0$ for any $2, \dots, s - 1$. We observe that $|u_1, v, u_s, u_j| / |u_1, v, u_s|$ is the signed distance of point u_j from plane (u_1, v, u_s) . Since u_j is above this plane, $|u_1, v, u_s, u_j| \geq 0$ implies $|u_1, v, u_s| > 0$ and establishes the lemma.

Therefore the previous process can be applied to σ' in the plane. Now the seam (referred to as 2-seam, for 2-dimensional seam) is formed by the vertices of σ' having a supporting line parallel to the y -axis. Specifically, $v \in \sigma'$ belongs

to the 2-seam if the mentioned supporting line leaves both edges incident on v in the same closed half-plane, and the interior of the upper edge in the open half-plane. We project all seam vertices to the x -axis and conclude that Γ is a convex polytope if and only if the latter projection consists of the boundary of a one-dimensional convex set, that is, it consists of exactly *two* points.

We observe that the test for membership of an edge in the seam is a subcomputation of its test for local convexity. Namely, let $\Delta_{1234} > 0$ be the condition for local convexity on edge (p_1p_2) (see Figure 2).

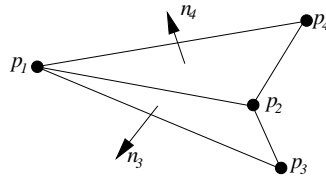


Fig. 2. The outward normals n_4 and n_3 .

The outward normal n_4 to $p_1p_2p_4$ has z -component

$$\begin{vmatrix} x_2 - x_1 & y_2 - y_1 \\ x_4 - x_1 & y_4 - y_1 \end{vmatrix} = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_4 & y_4 & 1 \end{vmatrix} = \Delta_{124}$$

and n_3 has z -component

$$-\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} = -\Delta_{123}$$

Therefore, if we evaluate Δ_{1234} by expansion according to the z -column, we obtain (for free) the minors required to test vertical support ($\text{sign}(\Delta_{124}) = \text{sign}(\Delta_{123})$).

Our T-checker for 3-dimensional convex polytopes is shown in Fig. 3. The certificate describing the topology is used in the local convexity test. We maintain two counters L and R , and, for every vertex v of Γ , a pointer $\text{seam}(v)$ to another vertex of Γ . Function $\text{seamvertex}(u, v, w)$ tests for seam membership in two dimensions and, depending upon whether v is not a seam vertex, is a left seam vertex, or is a right seam vertex, it returns 0, 1, or 2, respectively. Specifically, v is a right seam vertex either if both u and w are to its left, or if it lies above the vertex with the same abscissa while the other vertex lies to its left. A left seam vertex is analogously characterized.

The above approach can be generalized to higher dimensions. Let Γ be a simplicial polytope in d dimensions. A j -facet of Γ is a simplex defined by j linearly independent points; a conventional facet is a d -facet. Given points p_1, \dots, p_j we

```

L := 0; R := 0;
foreach vertex  $v$  do  $seam(v) := nil$ 
foreach edge  $e = pq$  do
  if  $e$  is not locally convex
  then “reject”
  else if  $e$  is a seam edge
    then if  $seam(p) = nil$ 
      then  $seam(p) := q$ 
      else  $b := seamvertex(q, p, seam(p))$ 
        if  $b = 1$  then  $L := L + 1$ 
    if  $seam(q) = nil$ 
      then  $seam(q) := p$ 
      else  $b := seamvertex(p, q, seam(q))$ 
        if  $b = 2$  then  $R := R + 1$ 

if  $L = 1$  and  $R = 1$ 
then “accept”
else “reject”

```

Fig. 3. T-checker for 3-dimensional convex polytopes.

let $[p_1 \dots p_j]$ denote the $j \times (d + 1)$ matrix whose i -th row is $(x_1^{(i)}, \dots, x_d^{(i)}, 1)$; $\|p_1 \dots p_j\|$ is the determinant of the leftmost j columns of $[p_1 \dots p_j]$.

The initial local convexity test applies to a $(d - 1)$ -facet. Specifically, let $(d - 1)$ -facet F be defined by p_1, \dots, p_{d-1} , indexed so that $\|p_1 \dots p_{d-1}\| > 0$. F is shared by two d -facets respectively defined by p_1, \dots, p_{d-1}, p_d and $p_1, \dots, p_{d-1}, p_{d+1}$, for two additionally independent points p_d and p_{d+1} . We say that F is *locally convex* if $\|p_1 \dots p_{d-1} p_d\| > 0$ and $\|p_1 \dots p_{d-1} p_{d+1}\| > 0$. We also say that f belongs to the d -seam if $\|p_1 \dots p_{d-1} p_d\| > 0$ and $\|p_1 \dots p_{d-1} p_{d+1}\| \leq 0$. A $(d - 1)$ -facet in the d -seam is easily shown to belong to a hyperplane parallel to the x_d -axis. If we project all d -seam $(d - 1)$ -facets along the x_d -axis we obtain a $(d - 1)$ -dimensional polytope Γ' , which can be shown to inherit the property of local convexity. The seam test for Γ' now involves membership of $(d - 2)$ -facets in a $(d - 1)$ -seam. Each $(d - 1)$ -facet F in the d -seam shares a $(d - 2)$ -facet with $(d - 1)$ adjacent $(d - 1)$ -facets, all belonging to the d -seam. Let F' be one such $(d - 2)$ -facet (for $d = 3$ F' is a point), which is visited twice as all d -seam facets are visited; through the obvious use of a pointer the two sharing $(d - 1)$ -facets are linked to each other, and F' can be tested for membership in the $(d - 1)$ -seam. The process proceeds recursively until the 1-seam is reached, at which level a counter is available for incrementing when appropriate.

The results presented in this section are summarized in the following theorem.

Theorem 5. *There exists an optimal T-checker for convex polytopes in any fixed dimension d that runs in linear time and has degree d .*

4 T-checkers for Planar Subdivisions

In this section we consider T-checkers for planar subdivisions that take as input a two-dimensional geometric graph Γ and the ordering Ψ of Γ , and verify whether Γ induces a planar subdivision satisfying a certain predicate (e.g., convex faces).

A building block of such a T-checker is an algorithm that tests whether an ordering Ψ of a graph G is planar. A linear-time algorithm for answering the question was given by Kirkpatrick in [14]. His algorithm considers the circuits induced by Ψ and adds to G a new vertex v_c for each induced circuit c and a new edge (v_c, w) for each vertex w of c . The resulting augmented graph G^* is planar if and only if the ordering Ψ is planar. Thus, the planarity of Ψ can be checked by running a planarity-testing algorithm (e.g., [13]) on G^* . Besides its theoretical interest, however, this algorithm may not be the most suited for practical applications, since the implementation of a linear-time planarity testing algorithm is complex and requires sophisticated data structures.

We show that there exists a much simpler solution to the problem of testing in linear time whether an ordering Ψ of a connected graph G is planar. Our algorithm follows from basic results in planarity theory [12]. Namely, we determine the circuits induced by Ψ and check whether their number C is equal to $E - V + 2$, where V and E are the number of vertices and edges of G , respectively. Clearly, this takes linear time.

Lemma 6. *Let Γ be a two-dimensional connected geometric graph with a planar ordering Ψ . If Γ is not planar (i.e., it has crossing edges), then at least one circuit of Γ induced by Ψ is not planar (i.e., it is a self-intersecting polygon).*

By Lemma 6, verifying whether a connected geometric graph Γ with a planar ordering is a planar can be reduced to testing separately whether the circuits induced by Ψ are simple (i.e., not self-intersecting) polygons. Lemma 6 also provides an alternative and shorter proof of a result of [9] (Lemma 4.5), as the following corollary shows.

Corollary 7. *Let Γ be a connected two-dimensional geometric graph. If the ordering Ψ of Γ is planar, all the internal circuits induced by Ψ are triangles, and the outer circuit induced by Ψ is a convex polygon, then Γ is planar and induces a triangulation.*

Observe, however, that Corollary 7 cannot be easily extended to other types of planar subdivisions. For example, Figure 4 shows a triconnected geometric graph (obtained by removing two edges from a triangulation) that has a planar ordering but is itself not planar.

Lemma 8. *A connected two-dimensional geometric graph Γ with ordering Ψ is planar and induces a triangulation if and only if: (i) ordering Ψ is planar; (ii) all the internal circuits induced by Ψ are triangles; and (iii) the outer circuit induced by Ψ is a convex polygon.*

The input to our T-checker for triangulations is a two-dimensional geometric graph Γ with n vertices and the ordering Ψ of Γ . The T-checker constructs the circuits induced by Ψ and then performs the following sequence of 6 simple

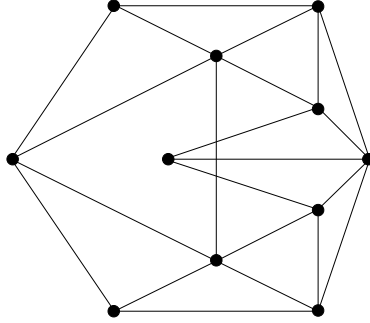


Fig. 4. A triconnected geometric graph whose ordering is planar, but which is itself not planar.

checks. It fails as soon as one of the checks fails, and succeeds if all the checks succeed. Checks 1 and 6 have degree 2. The other checks have degree 1.

1. Check that Ψ is the ordering of Γ .
2. Check that ordering Ψ is planar.
3. Check that Γ is connected.
4. Check that Γ has no more than $3n - 6$ vertices.
5. Check that all internal circuits induced by Ψ are triangles.
6. Check that the outer circuit induced by Ψ is a convex polygon.

Theorem 9. *There exists an optimal T -checker for planar triangulations that runs in linear time and has degree 2.*

Lemma 8 can be generalized to convex planar subdivisions.

Lemma 10. *A two-dimensional connected geometric graph Γ with ordering Ψ is planar and induces a convex planar subdivision if and only if: (i) ordering Ψ is planar; and (ii) all the circuits induced by Ψ are convex polygons.*

Theorem 11. *There exists an optimal T -checker for convex planar subdivisions that runs in linear time and has degree 2.*

As shown in Fig. 4, there exist nonplanar geometric graphs with a planar ordering and an underlying triconnected graph. In order to verify whether such geometric graphs are planar, the strategy suggested by Lemma 6 is to check the simplicity of the circuits induced by the ordering are simple polygons. Testing the simplicity of a polygon with k vertices can be done in $O(k \log k)$ by a simple optimal-degree sweep-line algorithm [3] or in $O(k)$ -time as an application of the more elaborate (and suboptimal in terms of degree) triangulation algorithm by Chazelle [6]. This discussion can be summarized as follows.

Lemma 12. *A connected geometric graph Γ with ordering Ψ is planar (and thus induces a planar subdivision) if and only if: (i) ordering Ψ is planar; and (ii) all the circuits induced by Ψ are simple polygons.*

Theorem 13. *There exists a T-checker for planar subdivisions that runs in linear time and has degree 3. Alternatively, there exists a T-checker that runs in $O(n \log n)$ time and has degree 2, where n is the number of vertices.*

5 N-checkers for Planar Subdivisions and Convex Polytopes

In this section, we present N-checkers for planar subdivisions. Our N-checkers compute the ordering of the input geometric graph and then use one of the T-checkers presented in Section 4.

Theorem 14. [14] *Let Γ be a geometric graph with n vertices such that its underlying graph G is planar and has $\lambda(G)$ distinct planar orderings. There exists an algorithm that either computes the ordering of Γ in $O(n + \log \lambda(G))$ time or fails. If it fails, then the ordering of Γ is not planar.*

Lemma 15. *The algorithm of Theorem 14 has degree 2.*

Our N-checker for planar triangulations exploits Theorem 14, Lemma 15, and the fact that the underlying graph G of a triangulation has $\lambda(G) = 2$. Let Γ be a geometric graph with n vertices. By Theorem 14, we can compute the ordering of Γ in $O(n)$ time or else we can conclude that Γ is not a triangulation (we reach this conclusion either when the running time of Kirkpatrick's algorithm exceeds $O(n)$ or when Kirkpatrick's algorithm fails in $O(n)$ -time). Once the ordering of Γ is computed, we apply the T-checker of Theorem 9.

Theorem 16. *There exists an optimal N-checker for planar triangulations that runs in linear time and has degree 2.*

Theorem 17. *There exists an N-checker for Delaunay triangulations that runs in linear time and has degree 4.*

Sketch of Proof. First, we verify that Γ is planar and induces a triangulation S using the T-checker of Theorem 16. To verify that S is a Delaunay triangulation, it suffices to check whether for every triangle $T = \Delta(a, b, c)$ of S , the disk through a, b, c contains any of the opposite vertices in the triangles sharing one edge with T . Clearly, this can be done in linear time. Also, the above in-circle test can be executed with a degree 4 algorithm (see, e.g. [16, 17]).

A *locally minimum-weight triangulation* is a triangulation such that for every edge shared by two triangles $\Delta(a, b, c)$ and $\Delta(a, b, d)$, edge bd is the shortest diagonal of the quadrilateral with vertices a, b, c, d . Locally minimum-weight triangulations have been extensively studied for their relationship to minimum-weight triangulations (see, e.g., [15]).

Theorem 18. *There exists an optimal N-checker for locally minimum-weight triangulations that runs in linear time and has degree 2.*

The optimal time complexity of the N-checkers of Theorems 16, 17, and 18 relies on the fact that $\lambda(G) = 2$ for the underlying graph of a triangulation.

The next lemma, shows that for convex planar subdivisions, $\lambda(G)$ is simply exponential.

Lemma 19. *Let G be the underlying graph of a convex planar subdivision. Then the number $\lambda(G)$ of topologically distinct planar orderings of G is $O(2^n)$, where n is the number of vertices of G .*

Sketch of Proof. The proof is based on the results of [7, 8], where planar graphs that admit a planar straight-line drawing with convex faces are characterized.

By combining Theorem 14, Lemma 15, Lemma 19, and Theorem 11 we obtain the following.

Theorem 20. *There exists an optimal N -checker for convex planar subdivisions that runs in linear time and has degree 2.*

Theorem 21. *There exists an N -checker for Delaunay diagrams that runs in linear time and has degree 4.*

Theorem 22. *There exists an optimal N -checker for three-dimensional convex polytopes that runs in linear time and has degree 3.*

6 Open Questions

Several questions remain open. Among them, we consider especially relevant the following.

1. Design effective T-checkers and N-checkers for other types of geometric graphs, such as Gabriel graphs, relative neighborhood graphs, and β -skeletons graphs.
2. Design a simple T-checker for planar subdivisions that runs in linear time and has degree 2.
3. Extend Theorem 22 on optimal N-checkers for 3-dimensional convex polytopes to any fixed dimension d .

Acknowledgments

We would like to thank Giuseppe Di Battista, David Kirkpatrick, Michael Goodrich, and Luca Vismara for useful discussions.

References

1. N. Amenta. Directory of computational geometry software. <http://www.geom.umn.edu/software/cglist/>.
2. G. Barequet, S. S. Bridgeman, C. A. Duncan, M. T. Goodrich, and R. Tamassia. Classic computational geometry with GeomNet. In *Proc. ACM Symp. Computational Geometry*, 1997.
3. J. L. Bentley and T. A. Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Trans. Comput.*, C-28:643–647, 1979.

4. M. Blum and S. Kannan. Designing programs that check their work. *J. ACM*, 42(1):269–291, Jan. 1995.
5. C. Burnikel. *Exact Computation of Voronoi Diagrams and Line Segment Intersections*. Ph.D thesis, Universität des Saarlandes, Mar. 1996.
6. B. Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6:485–524, 1991.
7. G. Di Battista, R. Tamassia, and L. Vismara. On-line convex planarity testing. In *Graph-Theoretic Concepts in Computer Science (Proc. WG 94)*, Lecture Notes in Computer Science. Springer-Verlag, 1994.
8. G. Di Battista, R. Tamassia, and L. Vismara. On-line convex planarity testing. Technical Report CS-95-26, Dept. Computer Science, Brown Univ., 1995. <ftp://ftp.cs.brown.edu/pub/techreports/95/cs95-26.ps.Z>.
9. G. Di Battista and L. Vismara. Angles of planar triangular graphs. *SIAM J. Discrete Math.*, 9(3):349–359, 1996.
10. I. Fary. On straight lines representation of planar graphs. *Acta Sci. Math. Szeged.*, 11:229–233, 1948.
11. S. Gottschalk, M. C. Lin, and D. Manocha. OBB-tree: A hierarchical structure for rapid interference detection. *Comput. Graph.*, 1996. Proc. SIGGRAPH '96.
12. F. Harary. *Graph Theory*. Addison-Wesley, Reading, MA, 1972.
13. J. Hopcroft and R. E. Tarjan. Efficient planarity testing. *J. ACM*, 21(4):549–568, 1974.
14. D. G. Kirkpatrick. Establishing order in planar subdivisions. *Discrete Comput. Geom.*, 3:267–280, 1988.
15. C. Levcopoulos and A. Lingas. On approximation behavior of the greedy triangulation for convex polygons. *Algorithmica*, 2:175–193, 1987.
16. G. Liotta, F. P. Preparata, and R. Tamassia. Robust proximity queries: an illustration of degree-driven algorithm design. *SIAM J. Computing*. to appear.
17. G. Liotta, F. P. Preparata, and R. Tamassia. Robust proximity queries: an illustration of degree-driven algorithm design. In *Proc. ACM Symp. Computational Geometry*, 1997.
18. K. Mehlhorn and S. Näher. *Checking Geometric Structures*, December 1996. Manual.
19. K. Mehlhorn, S. Näher, T. Schilz, S. Schirra, M. Seel, R. Seidel, and C. Uhrig. Checking geometric programs or verification of geometric structures. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 159–165, 1996.
20. K. Mehlhorn, S. Näher, T. Schilz, S. Schirra, M. Seel, R. Seidel, and C. Uhrig. Checking geometric programs or verification of geometric structures. Manuscript, 1997.
21. G. F. Sullivan, D. S. Wilson, and G. M. Masson. Certification of computational results. *IEEE Transactions on Computers*, 44(7):833–847, 1995.