Fast Algorithms for Generating Delaunay Interpolation Elements for Domain Decomposition

Philip L. Bowers
Department of Mathematics 3027
Florida State University
Tallahassee, Florida 32306-3027

William E. Dietz
Micro Craft, Inc.
207 Big Springs Avenue
P.O. Box 370
Tullahoma, TN 37388-0370

and

Stephen L. Keeling
Institut für Mathematik
Karl-Franzens-Universität Graz
Heinrichstraße 36
8010 Graz, Austria

Abstract

This work is motivated by the need to pass information efficiently among subdomains when solving partial differential equations with a domain decomposition approach. Because of ever greater demands for flexibility, no restrictive assumptions are made about the grids used to discretize the subdomains. Indeed, the union of all grid field points is treated as a general collection of points from which information must be interpolated onto a much smaller set of subdomain boundary points. A natural method of interpolation for a given boundary point involves identifying nearby field points as vertices of an encompassing Delaunay simplex, i.e., a simplex containing the boundary point while its circumsphere contains no field points. Accordingly, this paper presents methods for rapidly extracting these required interpolation elements from a Delaunay tesselation of field points without first constructing a much more costly global tesselation for the entire point set. The methods developed and analyzed have been termed the *Shrink Wrap* and *Oozing Bubble* methods. Proofs of convergence are provided for both methods. An example application from computational fluid dynamics is presented to demonstrate the use of these methods for hybrid computational grid systems.

For the research reported herein, the second two authors were supported in part while employed by Sverdrup Technology, Inc./AEDC Group, technical service contractor for the Arnold Engineering Development Center (AEDC), Arnold Air Force Base, TN 37389.

1 Introduction

This paper reports the development and analysis of fast methods for extracting selected elements from a Delaunay tesselation of a general set of points. The intent is for the selected elements to serve as interpolation stencils for another smaller set of points. Thus, for the applications of interest here, information is transferred from the larger to the smaller point set by local instead of global interpolation. No connectivities among any points are supplied, and the problem must be solved in much less time than that required to establish a global structure; therefore, computing a global Delaunay tesselation as a first step is prohibitive. The methods developed here are generally applicable to problems requiring interpolation of scattered data for a few, sparsely distributed points. Furthermore, the computational geometry results obtained contribute to the understanding of Delaunay tesselations, which are discussed in detail below.

This work is motivated by the need to pass information efficiently among subdomains when solving partial differential equations with a domain decomposition approach. For problems of interest here, the domain decomposition may be used simply to divide a very large problem into more manageable subproblems, or it may be used to facilitate multi-disciplinary analysis and region-specific model refinement. Thus, the subdomain equation sets may be identical or not. Furthermore, the subdomain grids may be contiguous or overlapping, static or dynamic, and structured or unstructured. In any case, information must be transferred among subdomains by interpolating values from subdomain field points to subdomain boundary points. A natural method of interpolation for a given boundary point involves identifying nearby field points as vertices of an encompassing Delaunay simplex, i.e., a simplex containing the boundary point while its circumsphere contains no field points. These simplices can be identified efficiently by quickly culling from the Delaunay tesselation of all subdomain field points, only those elements which cover the subdomain boundaries. These elements are in turn used as interpolation stencils for subdomain boundary points. This kind of information transfer capability is needed in a range of applications which are discussed in [9].

Such methods of information transfer have been applied in computational fluid dynamics (CFD) as follows [5], [3]. A typical CFD problem of interest to the U. S. Air Force is to simulate the trajectories of weapons released from a tactical aircraft configuration for the purpose of certifying safe separation. Such large and complex problems demand considerable flexibility in the solution procedure. For instance, the growing size of such problems means that domain decomposition will continue to be needed to achieve a partitioning into smaller parallel problems that do not exhaust available computer memory. Also, because of the great differences among flow regions, it can be necessary to employ different flow solvers to different regions depending upon the physical model and the grid type required. Finally, grid type constraints and restrictions on the connectivity among grids must be kept to a minimum. Specifically, subdomains must be simply shaped and allowed to overlap to facilitate rapid and easy grid generation by personnel working independently and in parallel. Even more importantly, subdomain overlap must be allowed so that a moving body and its conforming grids can be displaced together in the same rigid body motion; otherwise, a prohibitively expensive global regridding is required.

These flexibility requirements are met by the following domain decomposition approach. First, the field around the configuration is divided into simple subdomains with those near a solid body conforming to their respective portions of the configuration surface. Aside from the constraint that the union of these simple regions must cover the field, each subdomain is defined with little regard for the position of other subdomains. Therefore, one subregion may overlap another or actually penetrate a solid associated with a neighboring subregion.

After grids are generated for each of the subdomains, they are processed automatically in two ways. First, grid points that lie inside solid bodies are flagged and the points surrounding such flagged regions are identified to form new subdomain boundaries. Then every subdomain boundary is identified as either a physical boundary requiring physically appropriate boundary conditions or an artificial boundary requiring a link with some overlapping subdomain which provides interpolated boundary conditions. Thus, the global boundary value problem is replaced with a system of coupled boundary value problems posed on the respective subdomains.

Finally, the conservation laws are solved iteratively as follows. For a given time step, and a given subdomain, the discrete equations are solved using boundary values derived from fixed values on other subdomains. This step is performed for all subdomains. A subiteration can be used to enhance the quality of boundary values for the current time step. The process is repeated for a sequence of time steps until the final time is reached.

The step of primary concern in this paper is the rapid identification of interpolation stencils for artificial boundary points. For a given artificial boundary point, the interpolation stencil is formed as a Delaunay simplex with vertices taken from the complete collection of active field points for all subdomains. It is because of the required flexibility to use different flow solvers and different grid types on different subdomains that no connectivity information is assumed given for any part of the complete collection of points. Also, even though the artificial boundary points are few and sparsely distributed with respect to the complete set of field points, a significant number of interpolations may be required during a simulation. Therefore, the interpolation stencils must be identified as rapidly as possible.

Some investigators object to the use of such general overset grid methodologies because of concern about artificial boundary treatment violating a favorite notion of discrete conservation. Specifically, it is often thought that a numerical solution is critically flawed unless there is a global algebraic cancellation of cell interface fluxes. This of course is not a necessary condition for accuracy and the bias typically stems from a lack of experience with overset grid approaches. For clarification the reader is referred to [8] and the references cited therein.

The paper is organized as follows. Section 2 presents a quick review of Delaunay tesselations from the somewhat nontraditional vantage point of stereographic projection to a sphere. The advantage of this is that many issues of existence and uniqueness are clarified when viewed through the lens of stereographic projection. Two algorithms for fast extraction of Delaunay elements, respectively termed **Shrink Wrap** and **Oozing Bubble**, are examined in Sections 3 and 4. The proofs of convergence of these algorithms are greatly simplified by the use of stereographic projection followed by applications of elementary convex geometry. Furthermore, the two methods, when stereographically projected to a sphere, are surprisingly seen to be complementary versions of one and the same geometric pivoting algorithm with complementary initial data. In Section 5, the two algorithms are demonstrated in a CFD context along with additional details of implementation, including binning strategies. Here computational results are presented for a sample CFD problem. Finally, the paper closes with a summary and a discussion of future directions in Section 6.

2 Delaunay Tesselations and Stereographic Projections

In this section, Delaunay tesselations are defined and conditions are given for their existence and uniqueness. A useful and enlightening tool for understanding these tesselations is stereographic projection, which will be used later to analyze the Shrink Wrap and Oozing Bubble algorithms of Sections 3 and 4. These concepts are introduced in a general way for $\mathbb{R}^N$, but the particular interest is for $N = 2, 3$. For the following, the terms k-plane, k-sphere, k-ball, etc., denote k-dimensional planes, spheres, balls, etc. For instance, a 1-plane is a line, a 1-sphere is a circle, and a 2-ball is a disc. Also, $\mathrm{CH}(S)$ denotes the convex hull of the set S.

The interest of this paper is in a general (finite) collection of points in $\mathbb{R}^N$ with respect to which a Delaunay element is to be extracted that contains a fixed point of interest. To avoid certain trivialities and pathologies, this collection is assumed to contain at least $N+1$ points and to satisfy the following property.

Definition 2.1. A set of points $X \subset \mathbb{R}^N$ is said to be in **general position** if, for every integer k with $1 \le k \le N$, no set of $k+1$ points from X lies in a $(k-1)$-plane in $\mathbb{R}^N$.

Note that when X is in general position, every subset of X is also in general position.

Delaunay tesselations of a point set X are triangulations of the convex hull of X with certain desirable properties.

Definition 2.2. Given a set of points $X \subset \mathbb{R}^N$, a **Delaunay tesselation** $\mathcal{T}(X)$ is a set of N-simplices with the following properties:

i. for any $\sigma \in \mathcal{T}(X)$, the $(N-1)$-sphere circumscribing σ has no points of X inside,

ii. the collection of vertices among the N-simplices is equal to X,

iii. the elements of $\mathcal{T}(X)$ have disjoint interiors, and

iv. the N-simplices of $\mathcal{T}(X)$ cover $\mathrm{CH}(X)$.

Each N-simplex in $\mathcal{T}(X)$ is a **Delaunay simplex** and its companion circumscribing $(N-1)$-sphere is a **Delaunay sphere** determined by $\mathcal{T}(X)$.

It is well known that Delaunay tesselations exist for point sets in general position. We present a quick proof using stereographic projection to a sphere that shows, further, that the Delaunay tesselation is unique whenever X is nondegenerate in the following sense.

Definition 2.3. A set of points $X \subset \mathbb{R}^N$ is said to be **nondegenerate** if, for every integer k with $1 \le k \le N$, no set of $k+2$ points from X lies on a $(k-1)$-sphere in $\mathbb{R}^N$. Otherwise, X is said to be **degenerate**.

Degenerate point sets, as well as ones that fail to be in general position, are often encountered in practice, as in the case of points taken from a Cartesian grid. Nevertheless, it is straightforward to show that a point set X that suffers from one or both of these deficiencies can be perturbed by an arbitrarily small amount to produce a point set $\widetilde{X}$ that is both nondegenerate and in general position. Furthermore, the perturbation can be done in such a way that as $\widetilde{X}$ is continuously returned to X, the Delaunay tesselations, $\mathcal{T}(\widetilde{X})$, deform continuously to a Delaunay tesselation, $\mathcal{T}(X)$.

To get an intuitive sense of the existence of a Delaunay tesselation, consider the following. With $\mathbb{R}^N$ viewed as the subspace $\mathbb{R}^N \times \{0\}$ of $\mathbb{R}^{N+1}$, let the points of X be pushed ever so slightly to a set Y on a positively curved N-manifold $S \subset \mathbb{R}^{N+1}_{+} = \{y \in \mathbb{R}^{N+1} : y_{N+1} \geq 0\}$, where the subset of $\mathbb{R}^{N+1}_{+}$ above S is convex. Then the connectivities required for $\mathcal{T}(X)$ are revealed by the N-faces of $\mathrm{CH}(Y)$. This construction is made precise below with the use of stereographic projection, which provides a context in which the Shrink Wrap and Oozing Bubble methods easily can be seen to converge. Specifically, each scheme can visit only a finite number of states and, therefore, can fail to converge only if they cycle through some states. For both schemes, cycling is ruled out since stereographic projection reveals the existence of a function that is strictly monotone with respect to iteration. These issues will be discussed in detail in the next two sections. The framework is developed by recalling first the definition of stereographic projection.

Definition 2.4. Let $S^N = \{y \in \mathbb{R}^{N+1} : \|y\| = 1\}$ denote the unit N-sphere in $\mathbb{R}^{N+1}$ and $\Omega = (0, \ldots, 0, 1)$ the **north pole** of S^N. For $y \in S^N$, define a natural partitioning $y = (x, t)$ where $x \in \mathbb{R}^N$, $t \in \mathbb{R}$, and $\|x\|^2 + t^2 = 1$. Then the **stereographic projection** Π of a point $y = (x, t) \in S^N \backslash \{\Omega\}$ to $\mathbb{R}^N$ is defined by

$$\Pi(y) = \Pi(x, t) = \frac{x}{1-t} \in \mathbb{R}^N.$$

The **inverse stereographic projection** Π^{-1} of a point $x \in \mathbb{R}^N$ to S^N is given by

$$\Pi^{-1}(x) = \left(\frac{2x}{\|x\|^2 + 1}, \frac{\|x\|^2 - 1}{\|x\|^2 + 1} \right) \in S^N.$$

Stereographic projection, illustrated in Fig. 1 for a simple set of points in $\mathbb{R}^2$, can be defined

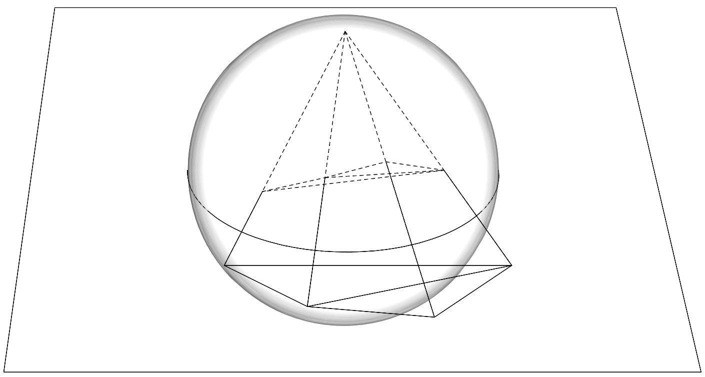

Figure 1: Illustration of Stereographic Projections for a Simple Delaunay Triangulation.

geometrically by the condition that for any $y \in S^N \backslash \{\Omega\}$, the points Ω, y, and $\Pi(y) \in \mathbb{R}^N = \mathbb{R}^N \times \{0\} \subset \mathbb{R}^{N+1}$ are collinear. Good references for the important properties of stereographic projection used below, and which are reviewed next, are [1] and [4].

Let $X \subset \mathbb{R}^N$ be a finite point set in general position with at least $N+1$ points and define $Y = \Pi^{-1}(X) \subset S^N$. Since all points of S^N are extreme points of their convex hull, $D = \mathrm{CH}(Y \cup \{\Omega\})$ is an $(N+1)$-dimensional polyhedron with vertex set $Y \cup \{\Omega\}$. This arrangement is illustrated in Fig. 1, where $N = 2$, X consists of 4 points in $\mathbb{R}^2$, Y consists of 4 points on S^2, and D is a hexahedron in $\mathbb{R}^3$.

One of the most important and useful properties of stereographic projection is that it and its inverse preserve spheres. Explicitly, if $S_x \subset \mathbb{R}^N$ and $S_y \subset S^N$ with $S_x = \Pi(S_y)$, then

1. S_x is an $(N-1)$-sphere if and only if S_y is an $(N-1)$-sphere with $\Omega \notin S_y$, and
2. S_x is an $(N-1)$-plane if and only if S_y is an $(N-1)$-sphere with $\Omega \in S_y$.

With $S^N \backslash S_y$ partitioned into two connected components, C_1 and C_2, the continuity of Π implies that $\Pi(C_1)$ and $\Pi(C_2)$ are connected components of $\mathbb{R}^N \backslash S_x$. Furthermore,

1. if $\Omega \in C_2$, then $\Pi(C_1)$ is the inside and $\Pi(C_2)$ the outside of the $(N-1)$-sphere, S_x, and
2. if $\Omega \in S_y$, $\Pi(C_1)$ and $\Pi(C_2)$ are half-spaces on opposite sides of the $(N-1)$-plane, S_x.

Examples of these facts are easily visualized when $N = 2$, where circles on the 2-sphere map to circles, except where a line in the plane is the projection of a circle containing the north pole.

Here is a quick argument using stereographic projection that proves the existence of Delaunay tesselations as well as clarifies the issues of nonuniqueness. Recall that a **support hyperplane** of an N-dimensional convex body C in $\mathbb{R}^N$ is an $(N-1)$-plane that meets C but does not disconnect C; in particular, C lies entirely in one of the closed half spaces determined by each of its support hyperplanes.

Theorem 2.1. *Let $X \subset \mathbb{R}^N$ be a finite point set in general position that contains at least $N+1$ points. Then there exists a Delaunay tesselation $\mathcal{T}(X)$. Moreover, the set X determines a unique collection of $(N-1)$-spheres in $\mathbb{R}^N$ that serves as the complete set of Delaunay spheres for* **every** *Delaunay tesselation for X. If X is nondegenerate, then the tesselation $\mathcal{T}(X)$ itself is unique.*

Proof. Let $Y = \Pi^{-1}(X)$. The convex hull of the set $Y \cup \{\Omega\}$ is an $(N+1)$-dimensional polyhedron D with vertex set $Y \cup \{\Omega\}$. Each N-dimensional face of D lies in a support hyperplane of D, an N-plane in $\mathbb{R}^{N+1}$ that intersects the unit sphere S^N in an $(N-1)$-sphere that contains the vertices of that face. Let $\mathcal{F}$ denote the collection of these $(N-1)$-spheres and partition $\mathcal{F}$ as $\mathcal{F}_\Omega \cup \mathcal{F}_0$, where $\mathcal{F}_\Omega$ is the set of those spheres passing through Ω while $\mathcal{F}_0$ is the set of those that miss Ω. The sphere-preserving properties of stereographic projection recalled above imply that, first, the Π-image of any sphere from $\mathcal{F}_\Omega$ is a support hyperplane for the convex hull $\mathrm{CH}(X)$, one for each of the $(N-1)$-dimensional faces of $\mathrm{CH}(X)$, and, second, the Π-image of any sphere from $\mathcal{F}_0$ is an $(N-1)$-sphere that does not contain any points of X in its interior. Let $\partial_0 H$ denote the union of the closed N-faces of ∂H that do not contain Ω. Finally, let $\mathcal{C} = \{C(S) : S \in \mathcal{F}_0\}$ where, for each S in $\mathcal{F}_0$, $C(S) = \mathrm{CH}(X \cap \Pi(S))$, the convex hull of the set of points of X that lie on the $(N-1)$-sphere $\Pi(S)$.

Notice that $\mathcal{C}$ is a collection of N-dimensional convex bodies, each of whose vertices lie on a sphere from $\Pi(\mathcal{F}_0) = \{\Pi(S) : S \in \mathcal{F}_0\}$ containing no points of X inside, and the collection of vertices among the elements of $\mathcal{C}$ is equal to X. Elementary linear algebra applied in $\mathbb{R}^{N+1}$ to illuminate stereographic projection to $\mathbb{R}^N$ verifies that for every y in $\partial_0 H$, there is a unique x

in CH(X) such that Ω, x, and y are collinear and, conversely, for every x in CH(X), there is a unique y in $\partial_0 H$ such that Ω, x, and y are collinear. From this it follows that the elements of $\mathcal{C}$, which may be described as the projections from Ω to $\mathbb{R}^N$ of the N-faces of D that lie in $\partial_0 H$, have disjoint interiors and that $\cup\{x \in C(S) : C(S) \in \mathcal{C}\}$ is exactly the convex hull CH(X). This shows that $\mathcal{C}$ satisfies the four defining properties of Delaunay tesselations of X from Definition 2.2, except that an element, $C(S) \in \mathcal{C}$, rather than necessarily being an N-simplex, may be an N-dimensional convex set spanned by more than $N+1$ points of X. Nevertheless, such a $C(S)$ may always be readily divided into N-simplices which in turn provide a local Delaunay tesselation for $C(S)$. Letting $\mathcal{T}(X)$ denote the set of all such N-simplices from these triangulations of elements of $\mathcal{C}$ produces the desired Delaunay tesselation.

One of the nice things about the existence argument above is that one may analyze exactly the nonuniqueness involved in forming Delaunay tesselations. Indeed, any $(N-1)$-sphere in $\mathbb{R}^N$ whose interior misses the point set X pulls back via Π^{-1} to an $(N-1)$-sphere on S^N that lies on an N-plane in $\mathbb{R}^{N+1}$ that does not disconnect D. In particular, if Σ is a Delaunay sphere from any Delaunay tesselation $\mathcal{T}$ of X, then its pull-back $S = \Pi^{-1}(\Sigma)$ lies on a support hyperplane for D that meets D in one of its N-dimensional faces. Therefore, S is an element of $\mathcal{F}_0$. This implies that each Delaunay sphere for $\mathcal{T}$ is contained in the set $\Pi(\mathcal{F}_0)$. This fact in turn implies that each element of $\mathcal{T}$ is contained in an element of $\mathcal{C}$. Since the elements of $\mathcal{C}$ have pairwise disjoint interiors, it follows, as the reader may verify, that every sphere in $\Pi(\mathcal{F}_0)$ is a Delaunay sphere for $\mathcal{T}$. Thus the set of Delaunay spheres for $\mathcal{T}$ is precisely the set $\Pi(\mathcal{F}_0)$, and the second claim of the theorem is verified. Finally, as the set of spheres $\Pi(\mathcal{F}_0)$ is uniquely determined by X, so too is the cell complex $\mathcal{C}$. It follows that the only nonuniqueness possible in forming Delaunay tesselations is in creating local Delaunay tesselations for the elements $C(S) \in \mathcal{C}$. There are in general several ways to accomplish such a subdivision; however, if X is nondegenerate, then each $C(S)$ is an N-simplex, and $\mathcal{C} = \mathcal{T}(X)$ is the unique Delaunay tesselation for X. $\square$

The lens of stereographic projection brings into focus not only the critical issues concerning existence and uniqueness of Delaunay tesselations, but also provides the primary tool that is used subsequently for confirming convergence of the algorithms presented in this paper. This geometric tool is encased in the next proposition.

Proposition 2.2. *Let S and S^* be $(N-1)$-spheres in $\mathbb{R}^N$ that meet in an $(N-2)$-sphere that lies in the $(N-1)$-plane E. Let H be an open half space of $\mathbb{R}^N$ bounded by E and assume that $S \cap H$ is contained in $B^* \cap H$, where B and B^* denote the respective interiors of S and S^*. Let J be the straight line segment in $\mathbb{R}^{N+1}$ with endpoints Ω and $\Pi^{-1}(x_0)$, where x_0 is any point in $B \cap H$, and let P and P^* be the N-planes in $\mathbb{R}^{N+1}$ containing the respective N-spheres $\Pi^{-1}(S)$ and $\Pi^{-1}(S^*)$. Finally, let y be the unique point of intersection of J and P, and y^* the unique point of intersection of J and P^*. Then y^* is strictly between Ω and y and, in particular, the distance from Ω to y is greater than that from Ω to y^*.*

Proof. Since x_0 is a convex combination of two points, say x_1 and x_2, in $S \cap H$, y is a convex combination of the points, $y_1 = \Pi^{-1}(x_1)$ and $y_2 = \Pi^{-1}(x_2)$, in P. Also, since $S \cap H$ is in B^*, the pre-image $\Pi^{-1}(S \cap H)$ lies both on P and on the component of $S^N \backslash \Pi^{-1}(S^*)$ that excludes Ω. Hence, $\Pi^{-1}(S \cap H)$ is on the side of P^* opposite Ω. Therefore, y_1 and y_2 are on the side of P^* opposite Ω. By convexity, so is y, and it follows that y^* is strictly between Ω and y. $\square$

3 The Shrink Wrap Method

It is an immediate consequence of the proof of existence given in Theorem 2.1 that any N-simplex spanned by $N+1$ points from X whose circumscribing sphere contains no points of X in its interior is automatically a Delaunay simplex for some Delaunay tesselation for X. Such a simplex is called a **Delaunay element** for X, and the goal of this paper is to develop efficient methods for extracting a Delaunay element from a general collection of points so that the element contains a specified **target point**. Since efficiency is of utmost importance, this is to be done without constructing a global Delaunay tesselation for X. The Shrink Wrap scheme for accomplishing this goal is detailed in this section. Here and in the next section, the target point, denoted as x_0, is assumed to satisfy the following property to avoid degeneracies that can occur among the points in $X \cup \{x_0\}$.

Definition 3.1. A target point x_0 will be called **generic** with respect to a point set X in $\mathbb{R}^N$ if the set $X \cup \{x_0\}$ is in general position.

Notice that the set of generic target points for a finite point set X in general position is an open dense subset of, and of full measure in, its convex hull $\mathrm{CH}(X)$. As such, any nongeneric target point may be displaced by an arbitrarily small amount to one that is generic.

3.1 Definition of the Scheme

The steps of the Shrink Wrap scheme are illustrated for dimension $N = 2$ in Fig. 2, and can

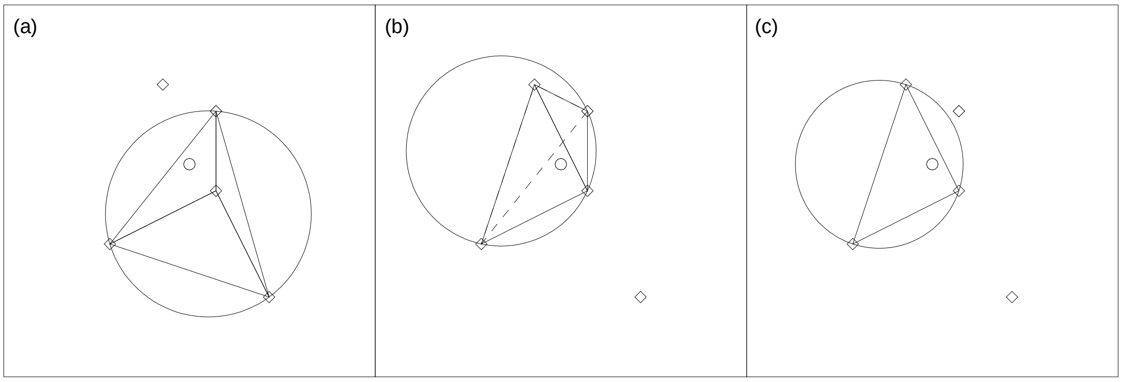

Figure 2: Illustration of the Shrink Wrap Scheme in $\mathbb{R}^2$. Grid point $= \diamond$. Target point $= \circ$.

be described briefly as follows. Assume that a given initial triangle, σ, has vertices from X and contains x_0 as is the case with the largest triangle in Fig. 2a. If the circle circumscribing σ has no points of X inside, as seen in Fig. 2c, then σ is the desired element. Otherwise, if there is a point $x \in X$ inside the circumcircle, as seen in Figs. 2a and 2b, then x is connected to the faces of σ to form subtriangles. Then σ is replaced with the subtriangle that contains x_0 and the procedure is repeated. A precise definition of the scheme is given subsequently.

For the following, let $S(\sigma)$ denote the $(N-1)$-sphere circumscribing an N-simplex σ and let $B(\sigma)$ denote the interior of $S(\sigma)$, the open N-ball bounded by $S(\sigma)$.

Algorithm 3.1. Shrink Wrap Algorithm.

A. Given: X in general position and a generic target point $x_0 \in \mathrm{CH}(X)$.

B. Given: $X_1 = \{x_{-j}\}_{j=1}^{N+1} \subset X$, where $x_0 \in \sigma_1 = \mathrm{CH}(X_1)$.

C. For $i = 1, 2, 3, \ldots$

1. *If $X \cap B(\sigma_i) = \emptyset$, then accept σ_i and stop.*
2. *Let x_i be any point in $X \cap B(\sigma_i)$.*
3. *Define $X_i^+ = X_i \cup \{x_i\}$.*
4. *Define the $(N-1)$-faces of σ_i by $\phi_j = \mathrm{CH}(X_i \backslash \{x_j\})$.*
5. *Form the subsimplices, $c_j = \mathrm{CH}(\{x_i\} \cup \phi_j)$.*
6. *If $x_0 \in c_k$, set $X_{i+1} = X_i^+ \backslash \{x_k\}$ and $\sigma_{i+1} = \mathrm{CH}(X_{i+1})$.*

Since X is in general position, each X_i is also in general position and each σ_i is an N-simplex. Similarly, X_i^+ is in general position and, as a result, the subsimplices are necessarily N-simplices. Since x_0 is generic, it cannot be on the boundary of any of the subsimplices and therefore must lie in the interior of exactly one c_k. If a nongeneric x_0 is encountered computationally, the subsimplex selected is the first one found to contain x_0. In any case, there is no ambiguity in the choice of x_k. On the other hand, there are options in the choice of x_i since $X \cap B(\sigma_i)$ may consist of more than one point, but the scheme converges for any choice of x_i.

3.2 Proof of Convergence

Since X is a finite set of points, there is only a finite number of simplices with vertices taken from X. Therefore, the Shrink Wrap scheme can fail to converge only by cycling through a finite collection of simplices. A natural strategy for proving that cycling cannot occur is to find some function depending on the simplices generated by the algorithm and not their indices while being strictly monotone with respect to iteration. Computational experience with the scheme generated an intuitive sense that something was being optimized in the process of iteration. However, it was only after considering the problem in the context of stereographic projection that an underlying basis for the intuition was revealed.

Theorem 3.1. *Let $X \subset \mathbb{R}^N$ be a finite point set in general position that contains at least $N+1$ points, and let x_0 be a generic target point. Then, in a finite number of iterations, Algorithm 3.1 identifies a Delaunay element for X that contains x_0.*

Proof. The algorithm can continue beyond step i only if $X \cap B(\sigma_i)$ is nonempty. Since X is a finite set of points, there is only a finite number of simplices with vertices taken from X and this means that the algorithm can fail to terminate only if $\sigma_i = \sigma_{i+l}$ for some indices i and $i+l$. This is ruled out as follows.

For any N-simplex σ in $\mathbb{R}^N$ whose circumscribing sphere, $S(\sigma)$, contains the target point x_0 in its interior, let $y(\sigma)$ denote the unique point of intersection in $\mathbb{R}^{N+1}$ of the N-plane containing $\Pi^{-1}(S(\sigma))$ with the line segment from the north pole Ω to x_0. Define $\delta(\sigma)$ as the distance from Ω to $y(\sigma)$.

Assume that x_0 lies in σ and let $\hat{x}$ be any point in $X \cap B(\sigma)$. Notice that since x_0 is generic, x_0 is contained in the simplex interior of σ, and there exists a unique $(N-1)$-face of σ whose vertices, along with $\widehat{x}$, span an N-simplex σ^* that contains x_0 in its interior. An application of Proposition 2.2 with $S = S(\sigma^*)$ and $S^* = S(\sigma)$ implies that $\delta(\sigma) < \delta(\sigma^*)$. Now let $\sigma = \sigma_i$, $\widehat{x} = x_i$ (from step C.2), and $\sigma^* = \sigma_{i+1}$ so that $\delta(\sigma_i) < \delta(\sigma_{i+1})$ for all i. In particular, it cannot happen that $\sigma_i = \sigma_{i+l}$ for integers i and $i + l$ since otherwise $\delta(\sigma_i) = \delta(\sigma_{i+l})$. $\square$

The intuitive picture provided by the proof, and which is quite transparent in the $N = 2$-dimensional case, is that the act of constructing σ_{i+1} from σ_i may be interpreted geometrically as "pivoting" an N-plane $P(\sigma_i)$ in $\mathbb{R}^{N+1}$ about an $(N-1)$-dimensional axis toward the projected target point $\Pi^{-1}(x_0)$. More will be said about this intuitive picture later when the Shrink Wrap and Oozing Bubble methods are compared.

4 The Oozing Bubble Method

The purpose of this section is to present another method for extracting a Delaunay element from a general collection of points so that the element contains a specified target point. The Shrink Wrap method generates a sequence of simplices that collapse around a Delaunay element containing the target point. On the other hand, the Oozing Bubble method generates a sequence of simplices that expand to meet a desired Delaunay element. As in the previous section, the target point x_0 is assumed to be generic with respect to the point set X.

4.1 Definition of the Scheme

The steps of the Oozing Bubble scheme are illustrated for dimension $N = 2$ in Fig. 3,

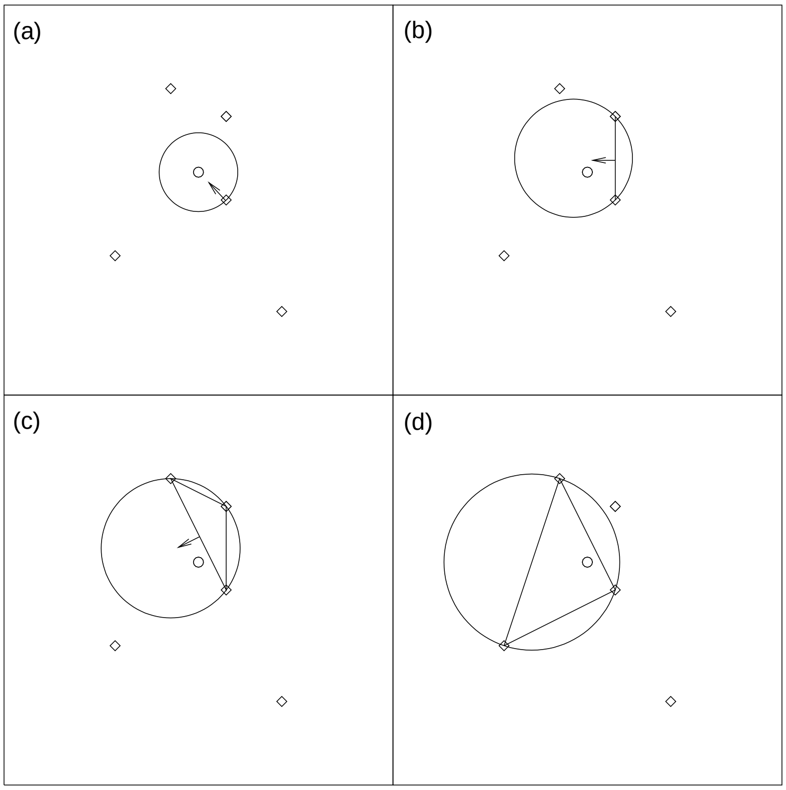

Figure 3: Illustration of the Oozing Bubble Scheme in $\mathbb{R}^2$. Grid point $= \diamond$. Target point $= \circ$.

and can be described briefly as follows. The scheme is started by finding the grid point closest to the target point as seen in Fig. 3a. Then the circle centered at the target point and passing through the nearest grid point is expanded in the space separating grid points until it meets points that form a Delaunay element. Specifically, the first expansion occurs by anchoring the circle at the first selected grid point, and then allowing it to swell in the direction of the target point until reaching another grid point as illustrated in Fig. 3a. The next expansion is directed orthogonally away from the current subspace of selected grid points and toward the side of the subspace on which the target point lies, as illustrated in Fig. 3b. Once enough simplex vertices have been found, they must span a Delaunay element since the expanding circle never surrounds any grid points. However, while the circle always contains the target point, the first Delaunay element found may not, as seen in Fig. 3c.

If the Delaunay element does contain the target point, as seen in Fig. 3d, it is the desired element so the scheme is halted. Otherwise, a vertex is discarded and the circle is pushed in another direction to find the correct element. The vertex to be dropped is determined as follows. The faces of the current simplex lie in subspaces which bound closed half-spaces that exclude the simplex interior. The first such half-space found to contain the target point excludes the vertex to be dropped. To find a replacement vertex, the current circle is expanded orthogonally away from the subspace of grid points retained and toward the side of the subspace on which the target point lies. This sequence of correction expansions continues until the required simplex is found. The precise description of the scheme is given next.

To set some notation for the following, whenever X_i^- is a k-element subset of $\mathbb{R}^N$ in general position, where $k \leq N$, let Q_i be the $(k-1)$-plane containing X_i^-, and let π_i denote the orthogonal projection onto Q_i. Also, for the vector n_i normal to Q_i, let D_i denote the open half space $\{x : n_i \cdot (x - x_i) > 0\}$.

Algorithm 4.1. Oozing Bubble Algorithm.

A. *Given: X in general position and a generic target point $x_0 \in \mathrm{CH}(X)$.*

B. *Find the point, $x_1 \in X$, closest to x_0 and set $a_1 = x_0$ and $X_1^- = \{x_1\}$.*

C. *For $i = 2, 3, \ldots, N$*

1. *Set $n_{i-1} = (I - \pi_{i-1})x_0$.*
2. *Set*
$$t_{i-1} = \min\left\{ \frac{\|x - a_{i-1}\|^2 - \|x_{i-1} - a_{i-1}\|^2}{2n_{i-1} \cdot (x - x_{i-1})} : x \in X \cap D_{i-1} \right\}$$
and let x_i be the point in $X \cap D_{i-1}$ where the minimum is achieved.
3. *Set $X_i^- = X_{i-1}^- \cup \{x_i\}$.*
4. *Set $a_i = a_{i-1} + t_{i-1}n_{i-1}$.*

D. *For $i = N+1, N+2, \ldots$*

1. *Set $n_{i-1} = (I - \pi_{i-1})x_0$.*
2. *Set*
$$t_{i-1} = \min\left\{ \frac{\|x - a_{i-1}\|^2 - \|x_{i-1} - a_{i-1}\|^2}{2n_{i-1} \cdot (x - x_{i-1})} : x \in X \cap D_{i-1} \right\}$$
and let x_i be the point in $X \cap D_{i-1}$ where the minimum is achieved.
3. *Set $X_i = X_{i-1}^- \cup \{x_i\}$.*
4. *If $x_0 \in \sigma_i = \mathrm{CH}(X_i)$, then accept σ_i and stop.*
5. *Define the orthogonal projections, $\tilde{\pi}_j$, onto the $(N-1)$-planes of $\mathrm{CH}(X_i \backslash \{x_j\})$.*
6. *If $(x_k - \tilde{\pi}_k x_k) \cdot (x_0 - \tilde{\pi}_k x_k) < 0$, set $X_i^- = X_i \backslash \{x_k\}$.*

7. *Set* $a_i = a_{i-1} + t_{i-1}n_{i-1}$.

Since x_0 is generic, $x_0 \notin Q_i$ and therefore n_i is always a nontrivial vector. Next, $X \cap D_i \neq \emptyset$ because of the following. First, $x_0 \in D_i$ since $n_i \cdot (x_0 - x_i) = n_i \cdot ([\pi_i x_0 + n_i] - x_i) = \|n_i\|^2 > 0$. So, if there were no points of X in D_i, this would be contrary to the assumption that $x_0 \in \mathrm{CH}(X)$. Since $X \cap D_i \neq \emptyset$ and $(x - x_i) \cdot n_i > 0$ for $x \in X \cap D_i$, t_i is well defined. If a nongeneric x_0 is encountered computationally, n_i is set to any vector that is normal to Q_i and that points into an open half space containing points of X. Then $X \cap D_i \neq \emptyset$ by construction.

Since X is in general position, X_i is also in general position and σ_i is an N-simplex. Note that there are options for the choice of x_i since the value of t_{i-1} can be achieved at more than one point in $X \cap D_{i-1}$. Though the scheme converges for any choice, assume for definiteness that x_i is the first minimizer in $X \cap D_{i-1}$ encountered.

For $i \geq N+1$, observe that $x_0 \in \sigma_i$ if and only if $(x_j - \tilde{\pi}_j x_j) \cdot (x_0 - \tilde{\pi}_j x_j) \geq 0$ for each $x_j \in X_i$. So, if $x_0 \notin \sigma_i$, there is at least one choice for point x_k in Step $D.6$ of the algorithm. However, there may be more than one since the inequality there may hold for more than one point in X_i. Though the scheme converges for any choice, assume for definiteness that x_k is the first point encountered in X_i for which the inequality holds. Note that x_k cannot be x_i because $x_i \in D_{i-1}$ and therefore, $(x_i - \tilde{\pi}_i x_i) \cdot (x_0 - \tilde{\pi}_i x_i) = (x_i - \pi_{i-1} x_i) \cdot n_{i-1} > 0$. As a result, x_i is always in X_i^-.

The next three lemmas verify that the algorithm has the characteristics illustrated in the description of the $N = 2$-dimensional case of Fig. 3. First, it is shown that for each i, the points of X_i^- all lie on an $(N-1)$-sphere centered at a_i. For the following, let $S(a; \bar{X})$ denote the $(N-1)$-sphere centered at a and passing through a set of points, $\bar{X}$.

Lemma 4.1. *For* $i \geq 1$, $X_i^- \subset S(a_i; \{x_i\})$, *and for* $i \geq N+1$, $X_i \subset S(a_i; \{x_i\})$.

Proof. The proof is by induction. First, let $i = 1$. Clearly, $X_1^- = \{x_1\} \subset S(a_1; \{x_1\})$. Now, suppose that $X_{i-1}^- \subset S(a_{i-1}; \{x_{i-1}\})$. Then, for each $x_j \in X_{i-1}^-$, $\|x_j - a_{i-1}\|^2 = \|x_{i-1} - a_{i-1}\|^2$. Also, since $x_{i-1} \in X_{i-1}^-$, $x_{i-1} - x_j \in Q_{i-1}$. Therefore, using the definition of a_i,

$$\|x_{i-1} - a_i\|^2 - \|x_j - a_i\|^2 = \|x_{i-1} - a_{i-1}\|^2 - \|x_j - a_{i-1}\|^2 - 2t_{i-1}n_{i-1} \cdot (x_{i-1} - x_j) = 0.$$

Next, by the definition of t_{i-1},

$$\|x_i - a_i\|^2 - \|x_{i-1} - a_i\|^2 = \|x_i - a_{i-1}\|^2 - \|x_{i-1} - a_{i-1}\|^2 - 2t_{i-1}n_{i-1} \cdot (x_i - x_{i-1}) = 0.$$

Combining the last two equations,

$$\|x_i - a_i\|^2 = \|x_j - a_i\|^2, \qquad x_j \in X_{i-1}^- \cup \{x_i\} = X_{i-1}^- \cup X_i^-$$

and $X_i^- \cup X_{i-1}^- \subset S(a_i; \{x_i\})$. Thus, the first statement of the lemma is established. Furthermore, this calculation shows that if $X_{i-1}^- \subset S(a_{i-1}; \{x_{i-1}\})$ then $X_i^- \cup X_{i-1}^- \subset S(a_i; \{x_i\})$. Since for $i \geq N+1$, $X_i^- \cup X_{i-1}^- = X_i$, the second statement of the lemma is obtained. □

For the following, let $B(\bar{a}; \bar{X})$ be the open N-ball bounded by $S(\bar{a}; \bar{X})$. Next it is shown that the N-balls, $B(a_i; X_i^-)$, never contain points of X.

Lemma 4.2. *For* $i \geq 1$, $t_i \geq 0$ *and* $B(a_i + tn_i; X_i^-) \cap X = \emptyset$ *for* $0 \leq t \leq t_i$.

Proof. The proof is by induction. First let $i = 1$. Then $X \cap B(a_1; X_1^-) = X \cap B(x_0; \{x_1\}) = \emptyset$, since x_1 is the point in X closest to x_0. Also,

$$t_1 = \frac{\|x_2 - x_0\|^2 - \|x_1 - x_0\|^2}{2n_1 \cdot (x_2 - x_1)}$$

is non-negative since $x_2 \in D_1$ and x_2 cannot be closer to x_0 than x_1. Now, assume that $t_{i-1} \geq 0$ and that $X \cap B(a_{i-1}; X_{i-1}^-) = \emptyset$. Suppose there is an $\hat{x} \in X \cap B(a_i; X_i^-)$. By Lemma 4.1, $x_{i-1} \in X_{i-1}^- \cup \{x_i\} \subset S(a_i; X_i^-)$, so

$$0 > \|\hat{x} - a_i\|^2 - \|x_{i-1} - a_i\|^2 = \|\hat{x} - a_{i-1}\|^2 - \|x_{i-1} - a_{i-1}\|^2 - 2t_{i-1}n_{i-1} \cdot (\hat{x} - x_{i-1}),$$

or

$$\|\hat{x} - a_{i-1}\|^2 - \|x_{i-1} - a_{i-1}\|^2 < 2t_{i-1}n_{i-1} \cdot (\hat{x} - x_{i-1}).$$

The left side must be non-negative since, by the inductive hypothesis, $X \cap B(a_{i-1}; X_{i-1}^-) = \emptyset$. Similarly, t_{i-1} is assumed to be non-negative. For strict inequality to hold, the given $\hat{x}$ must satisfy

$$0 \leq \frac{\|\hat{x} - a_{i-1}\|^2 - \|x_j - a_{i-1}\|^2}{2n_{i-1} \cdot (\hat{x} - x_j)} < t_{i-1}, \qquad x_j \in X_{i-1}^- \cap X_i^{-1}$$

which contradicts the definition of t_{i-1}. Therefore, $X \cap B(a_i; X_i^-) = \emptyset$. Furthermore,

$$t_i = \frac{\|x_{i+1} - a_i\|^2 - \|x_i - a_i\|^2}{2n_i \cdot (x_{i+1} - x_i)} \geq 0$$

since $x_{i+1} \in D_i$ and, as was just shown, $X \cap B(a_i; X_i^-) = \emptyset$.

Finally, suppose that, for some i, $t_i > 0$ and there is an $\hat{x} \in X \cap B(a_i + tn_i; X_i^-)$ for some $t \in (0, t_i)$. Then

$$\|\hat{x} - (a_i + tn_i)\|^2 < \|x_j - (a_i + tn_i)\|^2, \qquad x_j \in X_i^-$$

or

$$\|\hat{x} - a_i\|^2 - \|x_j - a_i\|^2 < 2t(\hat{x} - x_j) \cdot n_i, \qquad x_j \in X_i^-.$$

The left side must be non-negative since $X \cap B(a_i; X_i^-) = \emptyset$. Also, since $t > 0$, it must be that $n_i \cdot (\hat{x} - x_j) > 0$ for $x_j \in X_i^-$. Therefore,

$$0 \leq \frac{\|\hat{x} - a_i\|^2 - \|x_j - a_i\|^2}{2(\hat{x} - a_i) \cdot n_i} < t_i, \qquad x_j \in X_i^-.$$

However, this inequality contradicts the definition of t_i. As a result, $X \cap B(a_i + tn_i; X_i^-) = \emptyset$ for $0 \leq t \leq t_i$. □

Finally, it is shown that x_0 is always in $B(a_i; X_i^-)$.

Lemma 4.3. *For $i \geq 1$, $x_0 \in B(a_i; X_i^-)$.*

Proof. The proof is by induction. First let $i = 1$. Clearly, $x_0 \in B(a_1; X_1^-) = B(x_0; \{x_1\})$. Now, assume $x_0 \in B(a_{i-1}; X_{i-1}^-)$. Then, since $x_0 \in D_{i-1}$,

$$\|x_{i-1} - a_i\|^2 - \|x_0 - a_i\|^2 = \|x_{i-1} - a_{i-1}\|^2 - \|x_0 - a_{i-1}\|^2 + 2t_{i-1}(x_0 - x_{i-1}) \cdot n_{i-1} > 0.$$

By Lemma 4.1, $x_{i-1} \in X_i^- \cup X_{i-1}^- \subset S(a_i; X_i^-)$, so $x_0 \in B(a_i; X_i^-)$. □

4.2 Proof of Convergence

As with the Shrink Wrap method, the Oozing Bubble Algorithm can fail to converge only by cycling through a finite collection of simplices. Again, stereographic projection produces a function of the collection of simplices $\{\sigma_i\}$ that is strictly monotone with respect to iteration. The proof is very similar to that of Theorem 3.1.

Theorem 4.4. *Let $X \subset \mathbb{R}^N$ be a finite point set in general position with at least $N+1$ points, and let x_0 be a generic target point. Then, in a finite number of iterations, Algorithm 4.1 identifies a Delaunay element for X that contains x_0.*

Proof. Assume $i \geq N+1$. The algorithm can continue beyond step i only if x_0 is not an element of σ_i. Therefore, as in the proof of Theorem 3.1, the algorithm can fail to terminate only if $\sigma_i = \sigma_{i+l}$ for some indices i and $i+l$. This possibility is ruled out as in the proof of Theorem 3.1

For any N-simplex σ in $\mathbb{R}^N$ whose circumscribing sphere, $S(\sigma)$, contains the target point x_0 in its interior, let $y(\sigma)$ denote the unique point of intersection in $\mathbb{R}^{N+1}$ of the N-plane containing $\Pi^{-1}(S(\sigma))$ with the line segment from the north pole Ω to x_0. Define $\delta(\sigma)$ as the distance from Ω to $y(\sigma)$.

Assume that x_0 does not lie in σ. There is then a face τ of σ such that x_0 and the vertex of σ opposite τ lie on opposite sides of the $(N-1)$-plane E containing τ. Let H be the open half-space of $\mathbb{R}^N$ bounded by E that contains x_0 and let $\widehat{x}$ be any point of H that is outside the circumscribing sphere $B(\sigma)$. Let σ^* be the N-simplex spanned by $\widehat{x}$ and the vertices of τ, and note that x_0 is in the interior of the circumscribing sphere $S(\sigma^*)$. An application of Proposition 2.2 with $S = S(\sigma)$ and $S^* = S(\sigma^*)$ implies that $\delta(\sigma) > \delta(\sigma^*)$. Now, let $\sigma = \sigma_i$, $\widehat{x} = x_k$ (from step D.6), and $\sigma^* = \sigma_{i+1}$ so that $\delta(\sigma_i) > \delta(\sigma_{i+1})$ for all i. In particular, it cannot happen that $\sigma_i = \sigma_{i+l}$ for integers i and $i+l$ since otherwise $\delta(\sigma_i) = \delta(\sigma_{i+l})$. □

It should be clear from the proofs of convergence that the Shrink Wrap and Oozing Bubble Algorithms are two sides of the same coin. Viewed from the perspective of stereographic projection, Shrink Wrap begins with an N-plane P cutting through the convex body $D = \mathrm{CH}(\Pi^{-1}(X) \cup \{\Omega\})$ and separating Ω from the projected target point $y_0 = \Pi^{-1}(x_0)$. The plane P moves toward y_0 by pivoting about $(N-1)$-faces of D, all the while in contact with the segment J connecting the north pole to y_0. Each act of pivoting increases the length of the subsegment of J from the north pole to the point of intersection of J and P. Pivoting continues until P contains the N-face of D that meets J. This resting place of P cuts S^N in an $(N-1)$-sphere that projects via Π to a Delaunay sphere circumscribing a Delaunay element that contains x_0. On the other hand, Oozing Bubble begins with P tangent to S^N at y_0. The plane P moves toward the north pole until it hits D and becomes a support hyperplane for D. It then pivots, first along lower dimensional faces of D until it contains an N-face of D, then along $(N-1)$-faces until it contains the contiguous N-faces. Throughout its walk among the N-faces of D, stepping from face to contiguous face, P always moves as a support hyperplane for D. Always, P is in contact with J, and in each step decreases the length of the subsegment of J from the north pole to the point of intersection of J and P. The walk continues until P steps on the N-face of D that meets J, and a Delaunay element is found as with Shrink Wrap.

5 Application to CFD

The impetus behind the development of the Delaunay algorithms outlined in Sections 3 and 4 was to perform interpolations at the boundaries of domains within which physically relevant systems of partial differential equations are being solved. Specifically, the generalization of domain decomposition methods related to computational fluid dynamics (CFD) was the overarching goal. In this section, an application of the Delaunay algorithms in a CFD context is presented.

5.1 An Example Problem

A CFD solution on a realistic three-dimensional configuration provides an illustration of the use of the Delaunay algorithms in a domain-decomposition approach. Fig. 4 is a depiction of the configuration, which consists of a body, wings, tail, and sting support assembly. After extensive testing in wind tunnels, this configuration has a large associated experimental database, and is therefore valuable for CFD validation studies [6]. Computationally, the solid surfaces are represented by a system of five structured grids, while the outer portion of the domain is represented by a single large unstructured mesh. The mesh structure on the symmetry plane of the configuration is depicted in Fig. 5, which shows the inner structured region encompassed by an outer unstructured grid. A detail of Fig. 5 is presented in Fig. 6, which more clearly shows the overlapping interface between the outer unstructured mesh and the body-conforming structured grid adjacent to the solid surface of the configuration. The turbulent Navier-Stokes equations are solved on the structured grid. The specific flow solver used [10] is of a node-centered finite-difference type; therefore, flow variables and boundary conditions are stored at the mesh points. The unstructured-grid flow solver, however, is

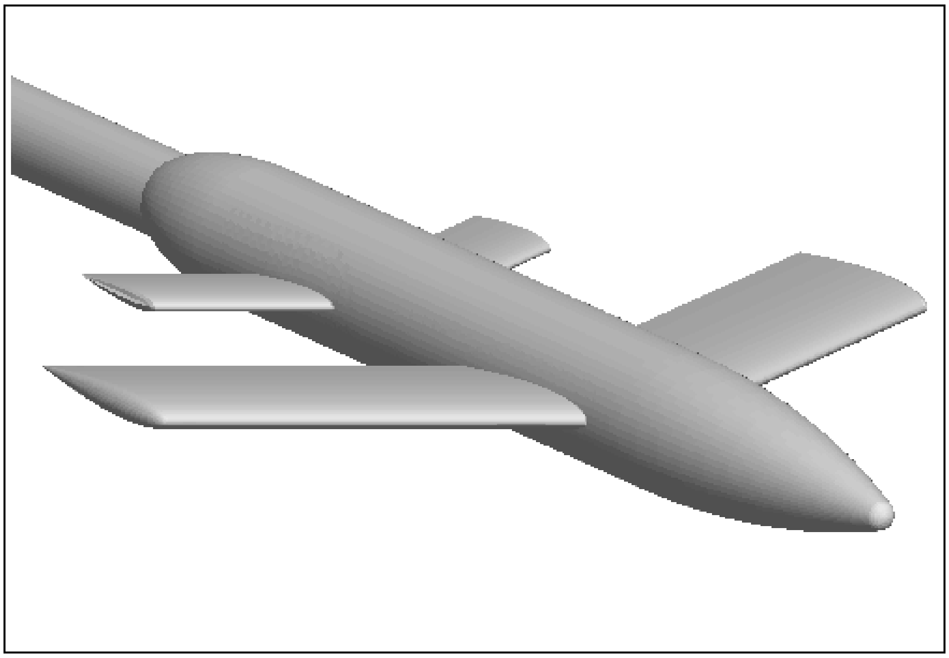

Figure 4: Wind Tunnel Wall Interference Model.

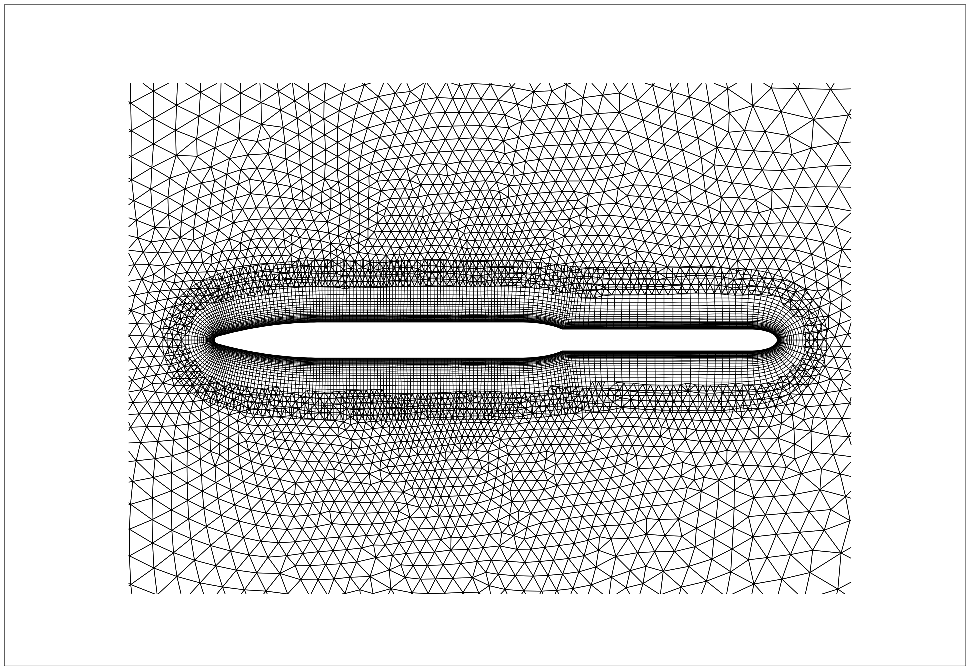

Figure 5: Symmetry Plane.

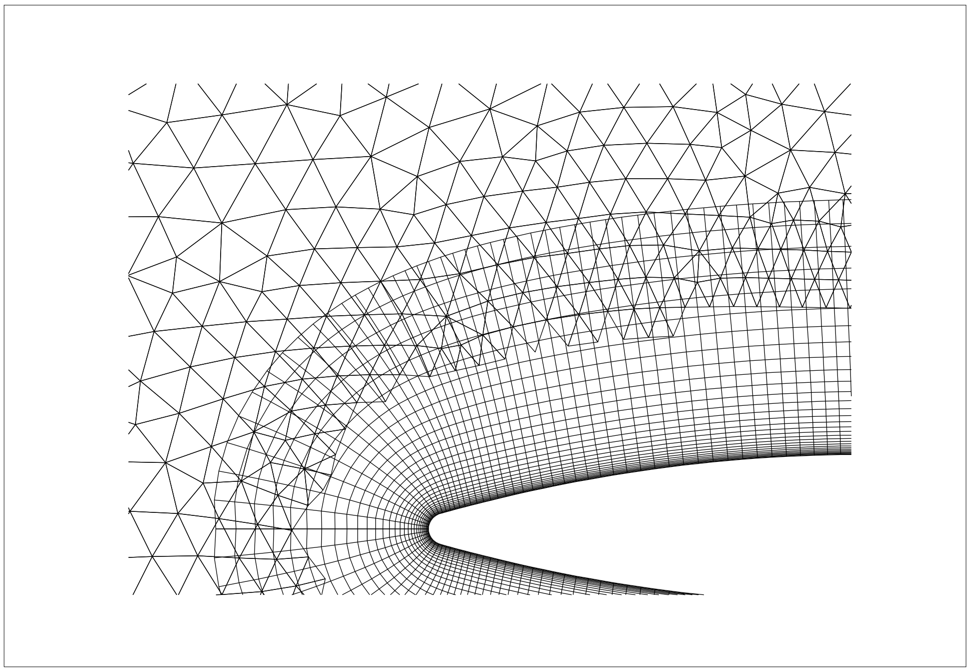

Figure 6: Symmetry Plane (detail).

based on a cell-centered finite-volume formulation [7]; its flow variables are stored at the centroids of the elements. The boundary conditions of the unstructured grid are specified at the centroids of the cell faces comprising the unstructured-grid boundary. In this study, the inviscid Euler equations were solved in the unstructured domain.

5.2 Previous Interpolation Methodology

Domain-decomposition methodologies, which rely on passing information between domains via interpolation at the boundaries, have been developed primarily for situations where complex geometries are represented by a system of structured grids, and where a uniform flow solver is used throughout the flow-field domain. In this standard approach, often referred to in the literature as the chimera method [2], the interpolation elements are specific hexahedra (3-D) within the meshes. The communication among the meshes in the system is established by identifying hexahedra in grids that overlap the boundary points and then updating boundary points through simple trilinear interpolation.

A domain decomposition methodology restricted to systems of structured grids limits the flexibility of approaches which can be used to solve various CFD problems. The use of unstructured, semi-structured, or hybrid grid structures would require special treatment. Consider, for example, the grid system presented in Figs. 5 and 6. In standard chimera methodology, hexahedral grid elements within the structured grid would provide interpolated flow-field variables to the boundary points of the unstructured grid, while tetrahedral elements in the unstructured grid would provide interpolated values to the boundary points of the structured grid. This approach would require knowledge of the different mesh structures, which meshes were supplying information to other meshes, and multiple specialized interpolation methods.

A further complication arises with the unstructured grid; the tetrahedral elements can be used as interpolation elements only if the flow-field values at the vertices are known. However, the flow variables in the unstructured grid are stored at the cell centroids. Therefore, flow variables at the cell vertices must be estimated in some manner. Various reconstruction methods may be applied; however, this adds an additional computational burden to the process. Simple average-based reconstruction methodologies tend to be highly dissipative, and would therefore be unsuitable for most compressible CFD applications.

These difficulties in extending the chimera methodology to other mesh structures and flow solver methodologies provided the main impetus for the development of a completely general interpolation methodology. The Delaunay algorithms presented in Section 3 and 4 form the heart of the interpolation methodology which is used in the generalized chimera scheme.

5.3 Present Interpolation Methodology

The boundary-condition interpolation methodology based on the Delaunay algorithms is illustrated in Figs. 7 through 9. The methodology is illustrated in two dimensions for clarity. The boundaries of the structured and unstructured grids are depicted in Fig. 7. The boundary of the structured grid is simply the locus of points on the outer surface of the structured grid; the boundary of the unstructured grid is depicted by the irregular boundary. Both boundaries are seen to be immersed in the calculated field points of both the structured and unstructured grids, which correspond to the grid points of the structured grid, and the centroids of the elements of the unstructured grid, respectively. Fig. 8 depicts the boundary points on each of the boundary surfaces, with the field

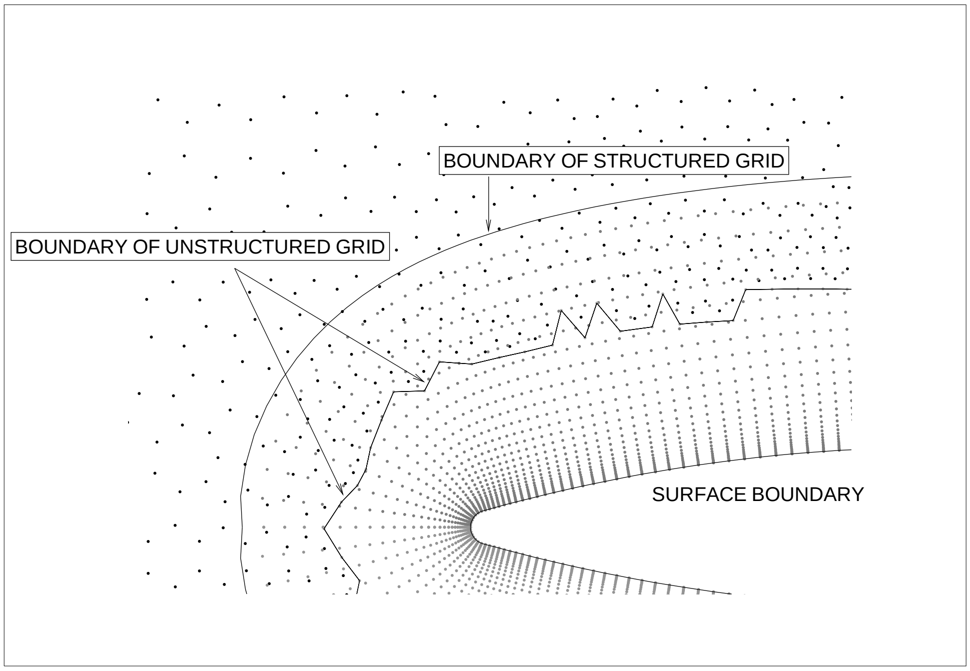

Figure 7: Boundaries of Structured and Unstructured Grids.

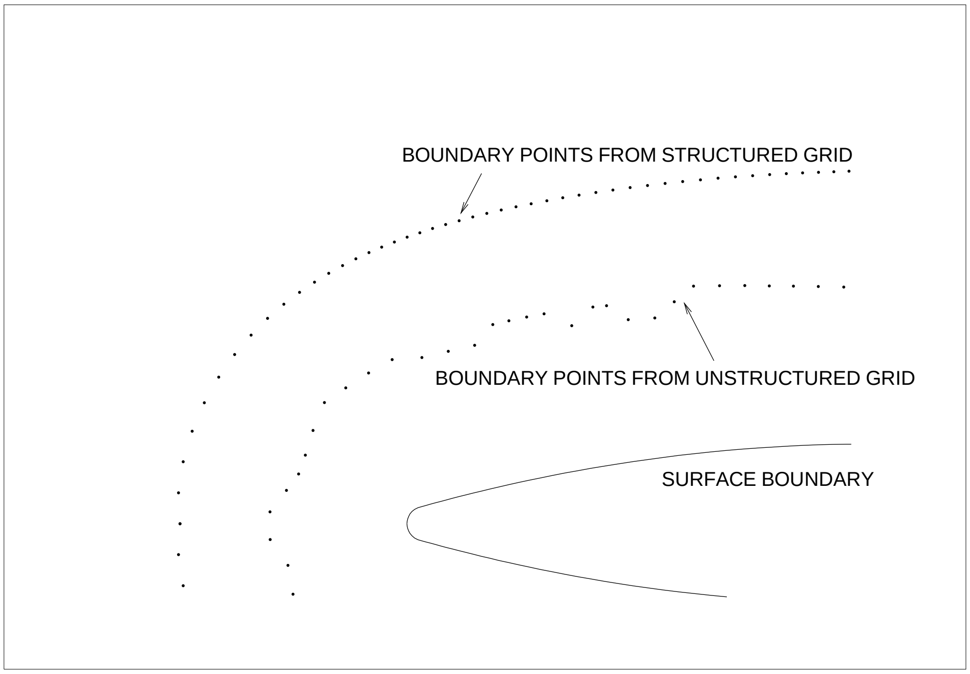

Figure 8: Boundary Points.

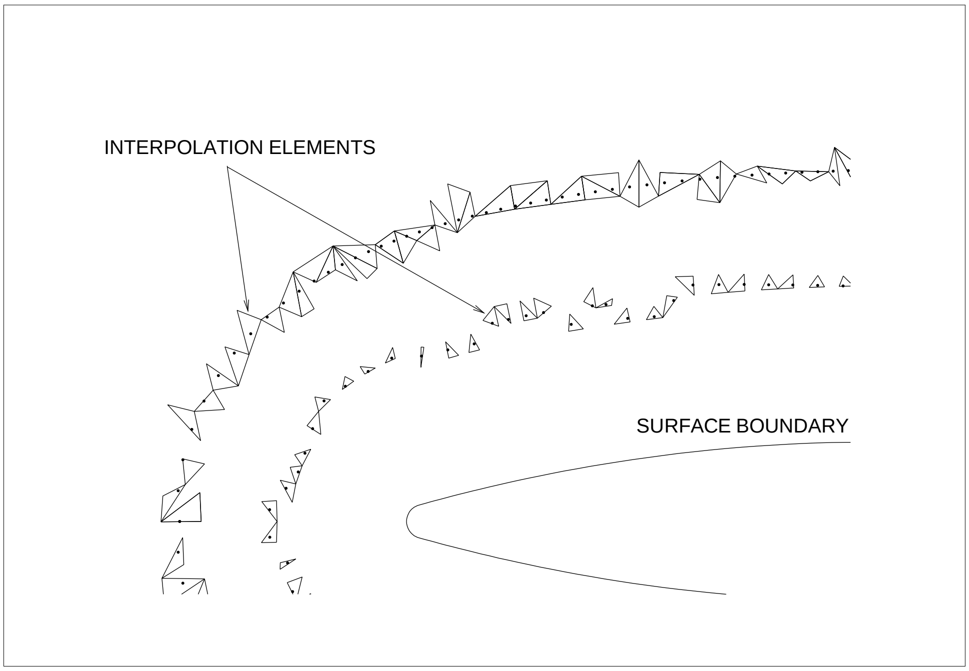

Figure 9: Interpolation Elements.

points removed for clarity. The boundary points of the structured grid correspond to the grid points of the outer boundary of the structured grid, while the boundary points of the unstructured grid occur at the midpoints of the faces on the boundary of the unstructured grid.

5.4 Restricting the Search

It would be possible to invoke either the Shrink Wrap or Oozing Bubble Delaunay methods at this point to generate interpolation elements about each boundary point. However, both methods require frequent interactions with potentially large point sets, for example, in calculating solutions to the general closest point problem, where the closest point to an arbitrary target point must be determined. Since a system of CFD grids representing a complex configuration can consist of millions of points, brute force exhaustive searches for closest points is computationally expensive, and would render the use of the Delaunay methods infeasible for most applications of interest.

In the present application, the Delaunay methods are invoked only for a region of points in the immediate neighborhood of each boundary point. The search restriction is accomplished by a preprocessing step. A Cartesian box, with edges aligned with the grid coordinate system, is placed around each mesh; each box is subdivided along each axis into a number of Cartesian sub-regions referred to as "bins". (A rather arbitrary division of the Cartesian region into 31 bins in each direction, for a total of 29791 bins, was used in the present study). Each field point in the mesh associated with the Cartesian box is assigned to a bin; a list of points each bin contains is then generated and stored. The bin in which an arbitrary target point resides can be determined quickly through a simple binary search in each coordinate direction.

Once the bin containing the target point is determined, the points in this bin and the immediately surrounding bins can be retrieved. At this juncture, the relatively small number of points retrieved may be searched sequentially to determine the closest point to the target point.

To be practical, a spatial searching method such as binning must be used before the Delaunay methods are invoked. In practice, simple binning as described above is not optimal for CFD applications because of the large discrepancies in mesh point density which are common to CFD problems. An adaptive binning procedure, such as octree searches, is preferable and is currently being implemented.

The Delaunay algorithms, either the Shrink Wrap or Oozing Bubble methods, can now be invoked within the region of each boundary point to develop unique Delaunay elements about the boundary points, where the vertices of each element are calculated field points from the structured and unstructured grids. The resulting elements are depicted in Fig. 9. Using the two-dimensional simplification for clarity, the elements are depicted as triangles; in the actual case, the elements are tetrahedra.

The generality of the interpolation approach is the crux of the generalized chimera methodology. The interpolation methodology is general across all mesh structures or flow solver formulations; no special cases need be accommodated. As a result, the methodology can support a variety of CFD domain decomposition approaches in a production environment. In the present implementation, simple linear algebraic interpolation on the elements is used to provide the necessary flow-field data at each boundary point. The flow solution procedure consists of two steps. The solution on each domain first is advanced one time step. Then the flow-field variables are updated at each boundary point. The steps are repeated until convergence is achieved (for a steady-state solution).

5.5 Computational Results

Fig. 10 depicts pressure contours on the symmetry plane of the configuration. Of particular interest is the behavior of the shock as it crosses the boundaries of the computational domains. It is known [8] that improper treatment of a shock at a boundary can cause shock displacement or spurious noise in the vicinity. No noise is evident in the region of the shock; other more benign features of the fluid flow-field also are consistent with proper interpolation at the boundaries. Double contour lines seen in some areas of the grid overlap region are common occurrences in chimera domain decomposition methods. They occur primarily due to different mesh resolutions in the overlap region, but may also be plotting artifacts stemming from different contour plotting algorithms used on the structured and unstructured domains.

Fig. 11 depicts computed pressure profiles on the body symmetry plane compared to experimental results. It is known that proper shock position will occur only if, in addition to interpolation being handled properly at the boundaries, the turbulent Navier-Stokes equations are solved in the region of solid surfaces. (Certain simplifications, such as the solution of the inviscid Euler equations, will result in shock displacement downstream, regardless of the integrity of the interpolation at the boundaries). It may be seen that the shock location is consistent with the experimental results, and compares well to the solution obtained with a fully-structured grid systems upon which the Navier-Stokes equations were solved on all grids. Fig. 12 depicts pressure profiles on the wing and tail surfaces at various span locations, compared to experimental results. Again, the shock location, highly sensitive to the integrity of the interpolation at the boundaries, compares well to experiment.

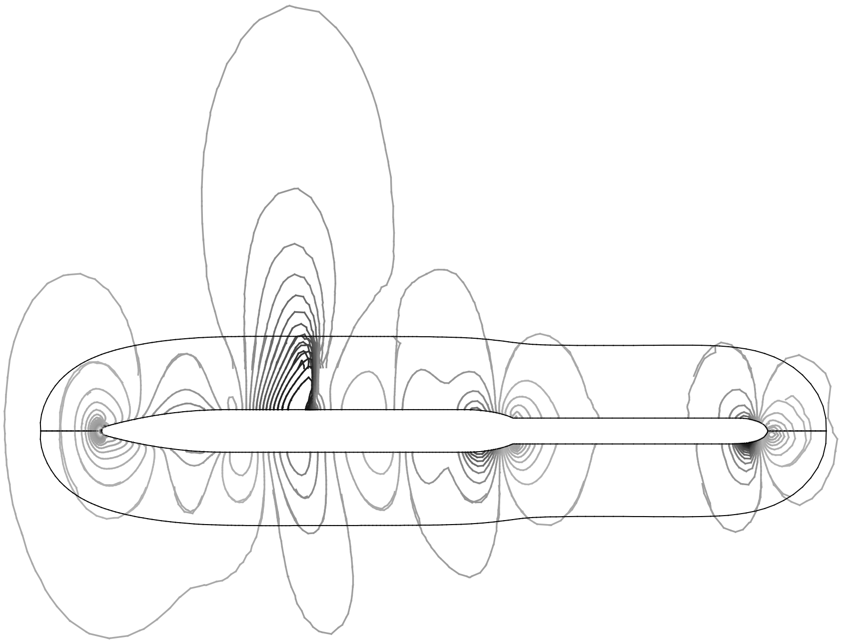

Figure 10: Pressure Contours on the Symmetry Plane.

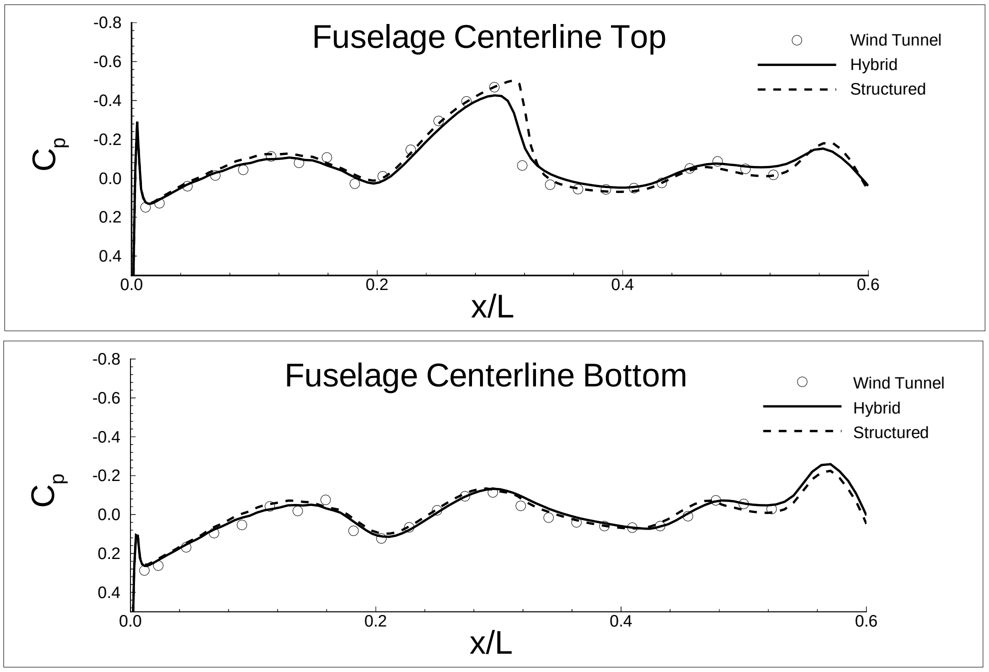

Figure 11: Pressure on the Symmetry Plane.

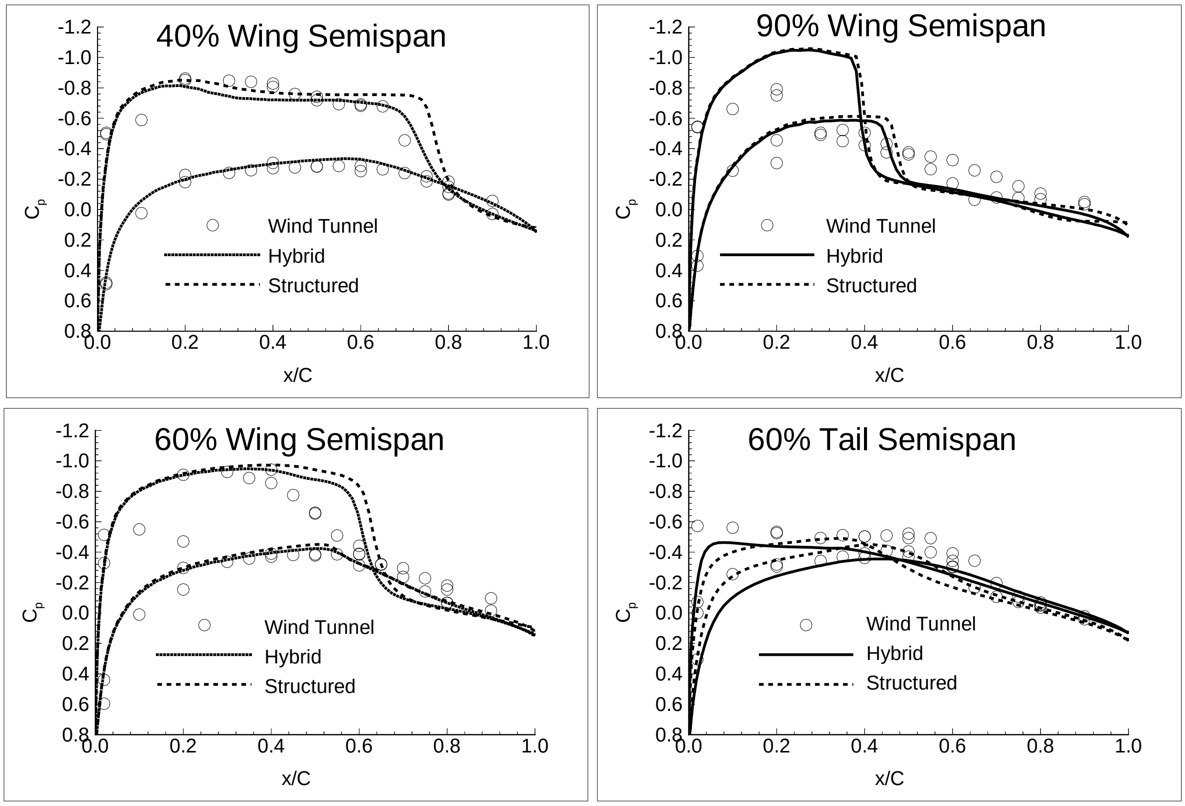

Figure 12: Pressure on the Wing and Tail Surfaces.

6 Conclusions

This paper, in general, has presented two methods for rapidly extracting selected elements from a Delaunay tesselation without first constructing the global tesselation. The particular interest here has been to use these methods as part of an efficient interpolation procedure for passing information among subdomains when solving partial differential equations with a domain decomposition approach. The specific task is to quickly cull from the the Delaunay tesselation of all subdomain field points, only those elements which cover the subdomain boundaries. These elements are in turn used as interpolation stencils for subdomain boundary points.

The two methods have been called the Shrink Wrap method and the Oozing Bubble method, and they both identify field points that form a Delaunay simplex around a given target point. Even though the two methods were developed independently, they were discovered to be complementary once viewed from the nontraditional perspective of stereographic projection. Specifically, it was found that connections among grid points, $X \subset \mathbb{R}^N$, required for a Delaunay tesselation are the same as the connections among the vertices, $Y = \Pi^{-1}(X) \subset \mathbb{R}^{N+1}$, of the convex hull, $D = \mathrm{CH}(\Omega \cup Y)$. Here, as earlier, Ω is the north pole of the N-sphere, $S^N \subset \mathbb{R}^{N+1}$, and Π is the stereographic projection of points on S^N to points in $\mathbb{R}^N$. On the one hand, the steps of the Shrink Wrap method were shown, through stereographic projection, to correspond to movements through D of an N-plane, P, which is constrained to pivot along $(N-1)$-faces of D until becoming tangent to D at a certain N-face. Then, vertices of this N-face project stereographically to the vertices of a Delaunay simplex encompassing the target point, x_0. It was shown that as P pivots, it divides the line between Ω and x_0 at points progressively further from Ω. On the other hand, the steps of the Oozing Bubble method were shown, through stereographic projection, to correspond to movements outside of D of an N-plane, Q, which is constrained to pivot along low-dimensional faces of D until becoming tangent to D at the same N-face described above. Also, it was shown in this case that as Q pivots, it divides the line between Ω and x_0 at points progressively closer to Ω. Since both schemes can visit only a finite number of states, they can fail to converge only if they cycle through some states. However, since the N-planes, P and Q, both move strictly monotonically between Ω and x_0, cycling is ruled out and convergence is thereby established.

An example application from computational fluid dynamics was presented to demonstrate the use of these methods for hybrid computational grid systems. For such practical implementations, it is important to note that both methods require some kind of treatment of possibly every single grid point, and consequently, a binning strategy must be used to limit the number of grid points involved in a given search. Furthermore, CFD problems typically involve meshes which vary greatly in point densities, so a simple binning approach could leave many bins nearly empty while others contain a large number of points. Therefore, the most current implementations involve an adaptive binning strategy, such as one incorporating an octree structure. Moreover, the latest implementations incorporate additional constraints to insure that each interpolation stencil has vertices representing the local mixture of grids and the most appropriate local grid spacing. Nevertheless, as ever more complex inter-grid communication schemes are designed, the two methods presented in this paper remain at the core of the more elaborate routines built around them.

References

[1] Beardon, Alan F., *The Geometry of Discrete Groups*, Springer-Verlag, New York, 1983.

[2] Benek, J. A., Steger, J. L., Dougherty, F. C., and Buning, P. G., *Chimera: A Grid-Embedding Technique*, AEDC-TR-85-64 (AD-A167466), December 1985.

[3] Benek, J. A., Steger, J. L., and Dougherty, F. C., *A Chimera Grid Embedding Technique with Application to the Euler Equations*, AIAA Paper No. 83-1944CP, July 1983.

[4] Berger, M., *Geometry I, II*, Universitext, Springer-Verlag, Berlin, 1987.

[5] Dietz, W. E., *Generalized Chimera, etc.*, Third Overset and Composite Grid and Solution Technology Symposium, Los Alamos, NM, November 18-21, 1996.

[6] Donegan, T. L., Benek, J. A., and Erickson, J. E., *Calculation of Transonic Wall Interference*, AAIA Paper No. 87-1432, June 1987.

[7] Frink, N. T., *Upwind Scheme for Solving the Euler Equations on Unstructured Tetrahedral Meshes*, AIAA Journal, Vol. 30, No. 1, January 1992.

[8] Keeling, S. L., Tramel, R. W., and Benek, J. A., *A Theoretical Framework for Chimera Domain Decomposition*, Advances in Flow Simulation Techniques–a Conference Dedicated to the Memory of Professor Joseph L. Steger, UC Davis, May 2-4, 1997. (Proceedings to be published by Springer-Verlag.)

[9] Löhner, R., *Robust, Vectorized Search Algorithms for Interpolation on Unstructured Grids*, J. Comp. Physics, Vol. 118, pp. 380-387, 1995.

[10] Tramel, R. W., and Nichols, R. H., *A Highly Efficient Numerical Method for Overset-Mesh Moving-Body Problems*, AIAA Paper No. 97-2040, June 1997.