ALGORITHMIC APSECTS OF ALTERNATING SUM OF VOLUMES, PART II. NON-CONVERGENCE AND ITS REMEDY

Kai Tang
Tony Woo

Department of
Industrial & Operations Engineering
University of Michigan
Ann Arbor, MI 48109-2117

Technical Report 90-30

October 1990

ALGORITHMIC ASPECTS OF ALTERNATING SUM OF VOLUMES

Part II.

Non-Convergence and its Remedy

Kai Tang
Tony Woo

University of Michigan

* Send correspondence to: Tony Woo
Dept. of Industrial & Operations Engineering
University of Michigan
Ann Arbor, MI 48109-2117
U.S.A.

Accepted by Journal of Computer Aided Design, to appear in March 1991.

Abstract

This is the second of a two-part paper. As the first part focused on the issues of data structure and fast difference operation, this part studies the non-convergence of the Alternating Sum of Volumes (ASV) process. An ASV is a series of convex components joined by alternating union and difference operations. It is desirable that an ASV series be finite. However, such is not always the case - that the ASV algorithm can be non-convergent. In this paper, the causes of this non-convergence are investigated and the conditions responsible for it is found and proven. Linear time algorithms are then developed for the detection.

1. INTRODUCTION

An Alternating Sum of Volumes (ASV) series is convergent if a deficiency Ω_n is the null set; otherwise, it is said to be non-convergent. (For computation of efficiency, the detection of a null deficiency Ω_n can be replaced by the determination of the convexity of Ω_{n-1}.) Figure 1 illustrates a non-convergent ASV series. The series of deficiencies Ω_1, Ω_2,..., as derived from the convex hull (CH) and difference (-) operations never converges to the null set, resulting in an infinite alternating series: $\{CH(\Omega) - CH(\Omega_1) + CH(\Omega_2) - \dots - CH(\Omega_{2i-1}) + CH(\Omega_{2i}) - \dots\}$.

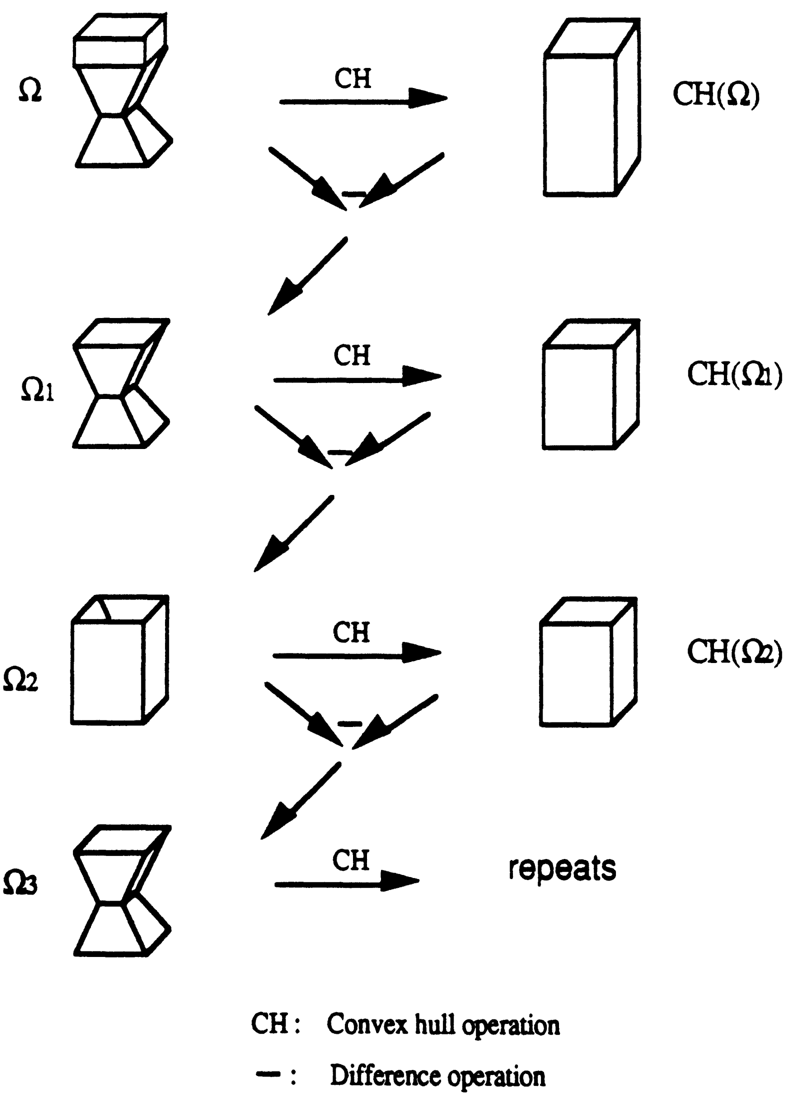


Figure 1. Illustration of ASV non-convergence

As implied in Figure 1, the non-convergence of an ASV series is determined by the non-convergence of a deficiency in its expansion. It is known [12] that an ASV series is non-convergent when the convex hull of a deficiency Ω_i identifies with the convex hull of the deficiency of Ω_{i+1}. For the example in Figure 1, the convex hull $CH(\Omega_1)$ is equal to the convex hull $CH(\Omega_2)$. As the result of the identification: $CH(\Omega_i)=CH(\Omega_{i+1})$, the following relation between the deficiencies holds: $\Omega_j=\Omega_{j+2}$ $(i\leq j)$.

Formally, a deficiency Ω_i is said to be *non-convergent* if the convex hull of its deficiency $CH(\Omega_i)-\Omega_i$ is equal to $CH(\Omega_i)$, and *convergent* otherwise. It is desirable to be able to characterize the non-convergence of a deficiency Ω_i directly, rather than invoking the comparision between $CH(\Omega_i)$ and $CH(\Omega_i)-\Omega_i$. This persuit is justified in two regards. First, four convex hull operations and two set difference operations must be performed to obtain the datum $CH(\Omega_i)$, $CH(\Omega_i)-\Omega_i$, and $CH(CH(\Omega_i)-\Omega_i)$ for the comparision. Set difference operation on a polyhedron with m vertices is known to take at least $O(m^2)$ time prior to the $O(m\log m)$ result in Part I of this paper. Secondly, even if the fast $O(m\log m)$ difference operation is involved, detecting the presence of a null set, as the result of the difference, can be numerically unstable. A fast non-convergence detection algorithm for a pseudo polyhedron without carrying on the set difference and comparision operations is offered as a new result for this part of the paper.

Suppose a fast non-convergence detection algorithm for a deficiency is available. One way to detect the non-convergence of an ASV series is to test for the non-convergence of every deficiency as it is being computed. The time required by such a detection scheme is heavily dependent on the depth n of the first non-convergent deficiency Ω_n -- the larger the number n is, the more time it will take. Alternatively, it may be inquired if the detection of the non-convergence of a series can be achieved without invoking the ASV process itself. Not only because the deficiencies thus produced are non-productive if that ASV series does not converge, but also because a separate scheme may speed up the detection time. From the theorectic point of view, such a study induces some interesting problems, such as that of finding the minimum number of faces in a non-convergent deficiency.

These two closely related issues, fast detection of the non-convergence of a deficiency and that of an ASV series, are investigated in this paper. In the next section, the concepts of *strong hull and weak hull vertices* are introduced. The characterization of these two types of vertices leads to an $O(n\log n)$ algorithm for detecting the non-convergence of a deficiency, where n is the number of vertices in the deficiency. In Section 3, a sufficient condition for the non-convergence of an ASV series is given, which requires only linear time to detect.

2. CHARACTERIZATION OF NON-CONVERGENT DEFICIENCIES

In this section, the following problem is to be solved: Given a pseudo polyhedron Ω_i, under what condition will the equation $CH(\Omega_i) = CH(CH(\Omega_i) - \Omega_i)$ hold and how fast can such a condition be detected? The symbols "CH" and "-" represent the convex hull and regularized difference operations, respectively. (Note that every deficiency in an ASV series must be a pseudo polyhedron, as shown in Part I of this paper. Hereafter, the two terms "pseudo polyhedron" and "deficiency" will be used interchangeably.) Before the condition for non-convergence is characterized, it is useful to summarize the relations between the boundary and interior points of a pseudo polyhedron Ω_i, its convex hull $CH(\Omega_i)$, and its deficiency $CH(\Omega_i) - \Omega_i$. The first relation, which has been shown in Part I of this paper, is re-cited below.

Lemma 1. The deficiency of a pseudo polyhedron Ω_i is also a pseudo polyhedron, whose interior $I(CH(\Omega_i) - \Omega_i)$ is the set difference $\{I(CH(\Omega_i)) - I(\Omega_i)\}$, and the boundary $B(CH(\Omega_i) - \Omega_i)$ is a subset of $\{B(CH(\Omega_i)) - B(\Omega_i)\}$ that forms the closure of $\{I(CH(\Omega_i)) - I(\Omega_i)\}$.

A pseudo polyhedron is completely described by its faces and a face is determined by its edges which themselves are defined by their end points called vertices. Since the vertices of the convex hull of a set of points must be a subset of that point set, by Lemma 1, the vertices of the deficiency of Ω_i is a subset of the vertices of Ω_i. In other words, the difference operation in the ASV expansion can be viewed as a *vertex elimination process*: After each difference operation, the deficiency Ω_i possesses fewer vertices than does the deficiency Ω_{i-1} ; this process continues until a convex pseudo polyhedron Ω_n is reached so that its deficiency Ω_{n+1} is the null set.

If the vertices in the deficiencies can not be eliminated through the difference operation, then the ASV series does not converge. A vertex of a pseudo polyhedron Ω_i is *eliminatable* if it does not exist in its deficiency $CH(\Omega_i)-\Omega_i$, otherwise it is *non-eliminatable*. A formal definition of the non-convergence of pseudo polyhedron is then in order.

Definition 1. A pseudo polyhedron Ω_i is non-convergent if all of its vertices are non-eliminatable; otherwise, it is convergent.

To characterize the eliminatability of vertices in Ω_i, the vertices are categorized into two groups, *hull vertices* and *internal vertices*. The hull vertices are those that are on the boundary of $CH(\Omega_i)$, whereas those vertices of Ω_i that are not on the boundary of $CH(\Omega_i)$ are internal. Each of the internal vertices has a three-dimensional neighborhood which is strictly inside $CH(\Omega_i)$. Furthermore, this neighborhood contains a subset of $\{I(CH(\Omega_i)) - I(\Omega_i)\}$ since an internal vertex is also a boundary point of Ω_i. Therefore, by Lemma 1 all the internal vertices are non-eliminatable. To study the eliminatability of the hull vertices, they are further separated into *weak* and *strong* hull vertices.

Definition 2. In E^3, the three-dimensional Euclidean space, a hull vertex of Ω_i is *weak* if it has a three-dimensional neighborhood that contains points in $\{\Omega_i \cup \{E^3 - CH(\Omega_i)\}\}$ only; otherwise it is called a *strong hull vertex*.

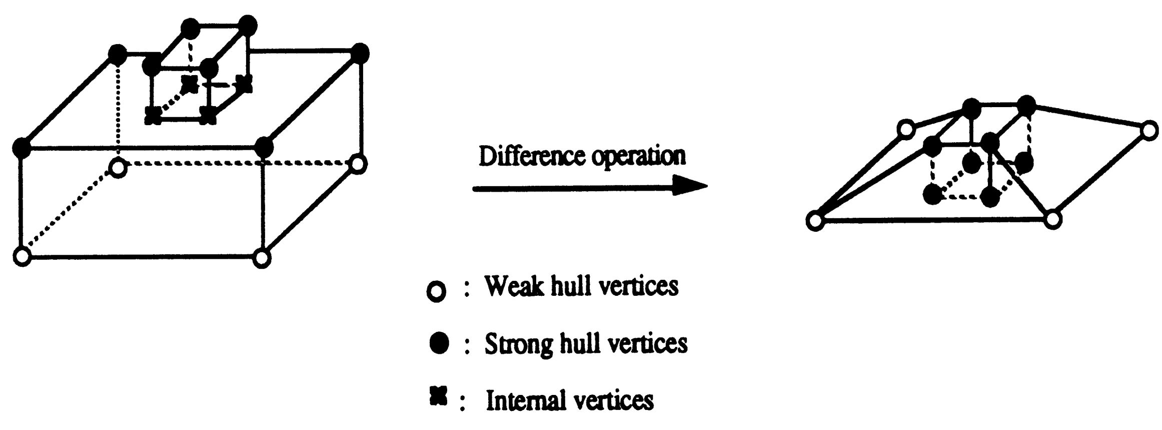


Figure 2. Weak, strong hull and internal vertices

As shown in Figure 2, after a difference operation, strong hull and internal vertices remain whereas all the weak hull vertices are eliminated. Let those faces (edges) of a pseudo polyhedron Ω_i be called *hull faces* (*hull edges*) if they are completely on the boundary surface of $CH(\Omega_i)$, and *internal faces* (*internal edges*) otherwise. Referring to Figure 2, it can be inferred that a hull vertex is weak if and only if all of its incident faces are hull faces of Ω_i.

(Note however that this condition does not hold for incident edges. That is, a hull vertex with incident hull edges only is not necessarily weak, as shown by Figure 3, where the strong hull vertex v has no incident internal edges.) The contribution of strong hull vertices to the non-convergence is manifested by the following lemma.

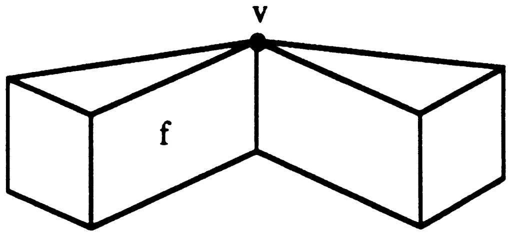


Figure 3. A strong hull vertex with no incident internal edges

Lemma 2. A pseudo polyhedron Ω_i is non-convergent if and only if all of its hull vertices are strong.

Proof. First it is noted that the hull and internal vertices partition the entire vertex set of Ω_i, due to their mutual exclusivity. By Definition 2, a weak hull vertex has an open three-dimensional neighborhood within which Ω_i is equal to $CH(\Omega_i)$ and thus there is no any subset of $\{I(CH(\Omega_i)) - I(\Omega_i)\}$ in that neighborhood. Hence, by Lemma 1, all the weak hull vertices are eliminatable. Conversely, since every three-dimensional neighborhood of a strong hull vertex contains a subset of $\{I(CH(\Omega_i)) - I(\Omega_i)\}$, they are preserved on the deficiency of Ω_i, i.e., they are non-eliminatable. By Definition 1 and the fact that all the internal vertices are non-eliminatable, the proof is complete.

Q.E.D.

Lemma 2 implies that the detection of the non-convergence of a pseudo polyhedron Ω_i is equivalent to distinguishing its strong hull vertices from the weak ones. Such a process takes two steps: classify the hull and internal faces of Ω_i, and then check if Ω_i has a vertex which has incident hull faces only. Whether a face is internal can be identified by way of checking that of one of its interior points. (Such a point must not be on an edge of the face since an internal face may have hull edges only, e.g. face f in Figure 3.) Ω_i is then non-convergent if and only if no weak hull vertex exists.

The algorithm given below follows the two steps just described. It is assumed that a procedure **HULL**(N,V,V_{tag}) is in hand, which takes a list V of N points as input and outputs a property array V_{tag} such that if V_{tag}(i) is "true" then point i in V is a hull vertex of CH(V); and "false" if it is an internal vertex.

Algorithm **DETECT** (Ω_i)

/* Detect the non-convergence of a pseudo polyhedron Ω_i.

The vertex list V and face list F of Ω_i have n_v vertices and n_f faces, respectively */

```
begin
step 1.   for k=1to nf do
              V(nv+k) <-- an interior point of face k in F
          end do

step 2.   call HULL(nv+nf,V,Vtag)

step 3.   set array VP(1: nv) to "true"

step 4.   for k=1 to nf do
              for every vertex v of face k in F do
                  VP(v) <-- VP(v)∩V(nv+k)
              end do
          end do

step 5.   for k=1 to nv do
              if VP(k)="true" then
                  return ('convergence')
              end if
          end do
```

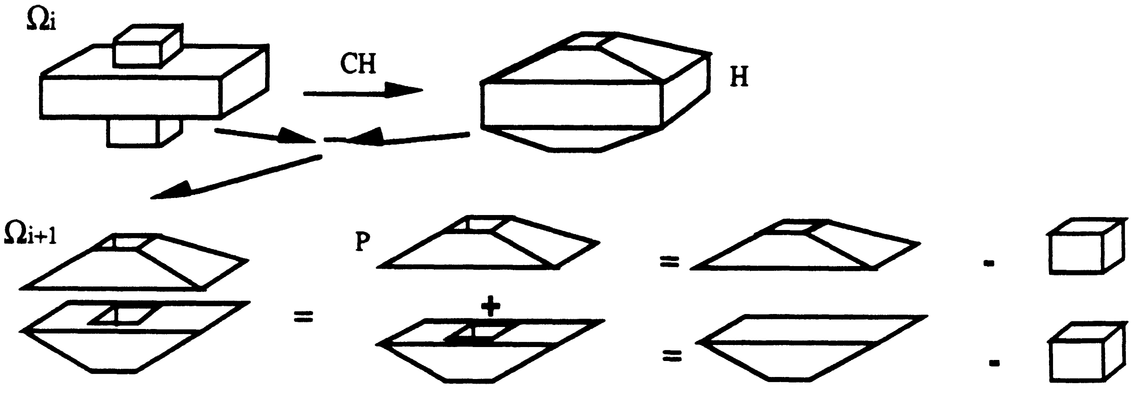
step 6. return ('non-convergence')
end **DETECT**

In the algorithm **DETECT**, the n_f interior points of the faces of Ω_i are first appended to the vertex array V of Ω_i. Since each interior point of a face can be obtained in constant time by considering any two adjacent edges of that face, step 1 takes $O(n_f)$ time. The convex hull procedure **HULL** is called at step 2 which requires only $O((n_v+n_f)\mathbf{log}(n_v+n_f))$ time [6]. At step 3, a property array $VP(1:n_v)$ is preset to "true". At step 4, the following is carried out: if a face k is internal, i.e., its interior point tag $V_{tag}(n_v+k)$ is "false", the corresponding entries in VP for all the vertices of face k are reset to "false". Such a process obviously takes O(D) time, where $D=\Sigma d_i$ $(i=1,2,...,n_v)$, and d_i is the degree of vertex i. It is shown in the Appendix of Part I of this paper that D is $O(n_f)$. Finally, at step 5, the array VP is scanned and Ω_i is identified as convergent if some entry in VP is "true", and non-convergent otherwise. The time complexity of the algorithm **DETECT** is summarized by the following theorem.

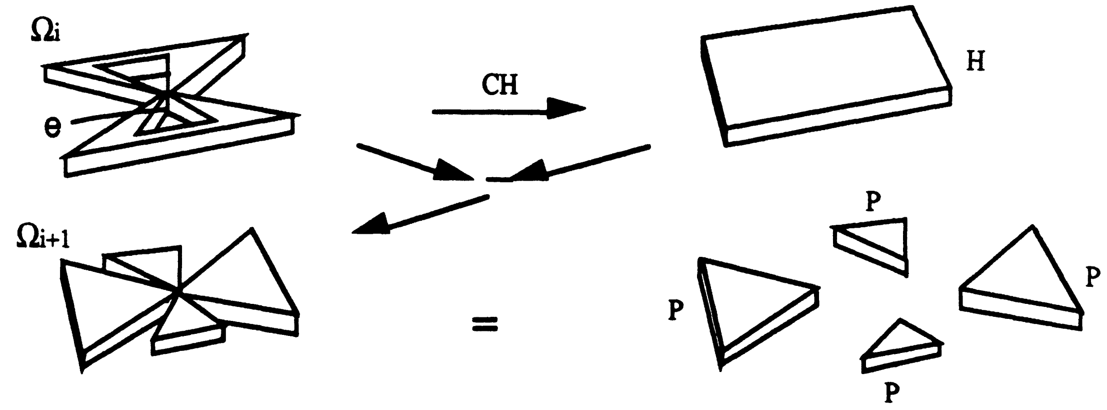
Theorem 1. The detection of the non-convergence of a pseudo polyhedron Ω_i with n vertices can be done in O(n**log**n) time.

Compared to the simple comparision method: $CH(\Omega_i) = CH(CH(\Omega_i)-\Omega_i)$ [12], the new detection algorithm **DETECT** avoids both the time consuming difference operation and the identification of a null set which could be numerically unstable. Two convex hull operations are also saved.

It may be noted that the detection algorithm **DETECT** disregards the disconnectedness of a set. The pseudo polyhedron Ω_i in Figure 4(a) is non-convergent by Lemma 2. The deficiency Ω_{i+1}, however, consists of two separate pseudo polyhedra P_1 and P_2. Though Ω_{i+1} is non-convergent as a single set, it is convergent if represented as $ASV(\Omega_{i+1}) = ASV(P_1+P_2) = ASV(P_1) + ASV(P_2)$ because P_1 and P_2 are both convergent. It results in a convergent ASV tree $ASV(\Omega_i) = H - \Omega_{i+1} = H - ASV(\Omega_{i+1}) = H - (ASV(P_1) + ASV(P_2))$, which branches at the deficiency Ω_{i+1}. In some other cases, a pseudo polyhedron, though connected, might be separated uniquely at some edges such that the separated subsets are all convergent. For example, the pseudo polyhedron Ω_i in Figure 4(b) is non-convergent by Lemma 2, it is however convergent if expressed as $\Omega_i = H - \Omega_{i+1} = H - (P_1 + P_2 + P_3 + P_4)$ since all P_1, P_2, P_3 and P_4 are convergent.



(a)



(b)

Figure 4. Convergence by set separation

Both examples of set separation on Ω_{i+1} shown in Figure 4 bear a crucial property: the boundary of the separated pseudo polyhedron remains unchanged. Unlike polyhedral decomposition [4], such a property guarantees that the boundary after the separation will have the same sets of vertices, edges, and faces, with only the adjacency and incidence relations among them altered. Furthermore, it will be shown next that such a separation is unique, thus justifying the existence of a deterministic algorithm. To define set separation rigrously, the concept of *well-connectedness* is needed.

Definition 3. Two points p and q of a pseudo polyhedron Ω_i are said to be *well connected* in Ω_i if there exists a curve c between p and q such that all the points in c, except for possibly p and q, are in $I(\Omega_i)$. Ω_i is a *well-connected set* if all of its points are well connected in it, otherwise it is a *ill-connected* set. (Refer to Figure 5.)

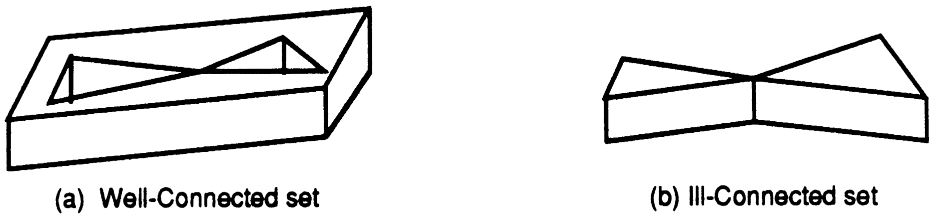


Figure 5. Well-connected set vs. Ill-connected set

The Ω_{i+1} in Figure 4(a) and both Ω_i and Ω_{i+1} in Figure 4(b) are ill-connected pseudo polyhedra. A well connected pseudo polyhedron is also called a *robust set*, meaning its interior is all connected. A subset ζ of a pseudo polyhedron Ω_i is a *maximally well-connected set* (MWCS) of Ω_i if ζ is a well-connected set and any addition of non-ζ points of Ω_i to ζ will constitute an ill-connected set. As an example, only P_1, P_2, P_3 and P_4 are the MWCSs of the pseudo polyhedron Ω_{i+1} in Figure 4(b).

It is desirable that an ASV series be expanded as much as possible so that more features can be extracted. Once a non-convergent and ill-connected deficiency is encountered, it should be separated into the MWCSs and the ASV process can then be performed on each of them. This leads to the notion of *strong* and *weak* non-convergence.

Definition 4. A non-convergent pseudo polyhedron Ω_i is *strongly non-convergent* if both itself and its deficiency are robust. Otherwise, Ω_i is *weakly non-convergent.*

As examples, the deficiency Ω_1 in Figure 1 is strongly non-convergent since both itself and its deficiency Ω_2 are robust, whereas each Ω_i in Figure 4 is weakly non-convergent because either itself or its deficiency Ω_{i+1} is ill-connected.

The detection of the strength of non-convergence of a pseudo polyhedron Ω_i of n faces requires three steps: the identification of its non-convergence, the computation for the deficiency of Ω_i, and the classification of the well-connectedness of Ω_i and/or its deficiency. The first step can be done, by Theorem 1, in O(n**log**n) time. The difference operation also requires O(n**log**n) time as shown in Part I of this paper. Thus, if the classification of the robustness of Ω_i can be done in O(n**log**n) time, so can the detection of the strong non-convergence. Since Ω_i is robust if and only if its MWCS separation contains only one MWCS, i.e., itself, the goal becomes the finding of an O(n**log**n) time MWCS separation algorithm

In implementing such an algorithm, it is noted that, by definition of a pseudo polyhedron, *the well-connectedness of its boundary point set will ensure it to be a well-connected set* . This fact ensures that the MWCSs of a pseudo polyhedron can be detected by checking only the well-connectedness of its faces.

Let two faces of a pseudo polyhedron be *well-adjacent* to each other if they share a common edge and are well-connected to each other. It can be easily shown that two faces A and B of a pseudo polyhedron are well-connected if and only if either they are well-adjacent or there exist a number of faces $f_1, f_2, ..., f_k$ such that A is well-adjacent to f_1, f_1 is well-adjacent to $f_2, \ldots$, and f_k is well-adjacent to B. For example, in Figure 6, faces A and B are not well-connected because the curve c connecting points p and q passes through edge "e" , which does not belong to the interior of that pseudo polyhedron.

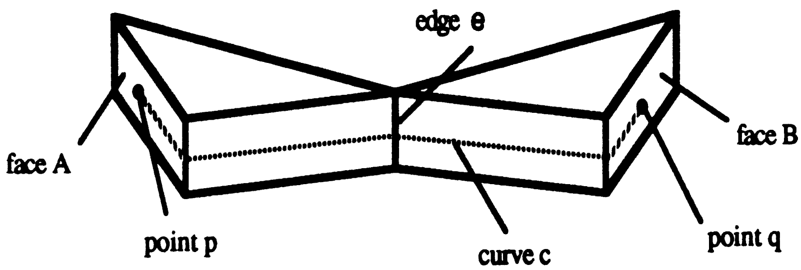


Figure 6. Ill-connectedness of faces of a pseudo polyhedron

To characterize face well-adjacency, let $f_1, f_2, ..., f_m$ be the faces incident to a common edge, ordered by their spatial angles. (In Figure 7, $f_1, f_2, ..., f_m$ are the intersections between the faces sharing a common edge and a plane orthogonal to that edge.) Apparently, the well-adjacent face of face f_i (i=1,2,...,m) is either f_{i-1} or f_{i+1} (mod m), depending on the direction of the outward normal of f_i. Such a pairing process can be done in O(m) time. The following recursive procedure **MWCS_FACES** finds the faces of a maximally well-connected set of a pseudo polyhedron Ω_i. The input is the pseudo polyhedron representation {V,E,F,NORM,E_f} of Ω_i and the index of a face of Ω_i. The output are the indices of those faces of Ω_i that form the boundary of a MWCS of Ω_i.

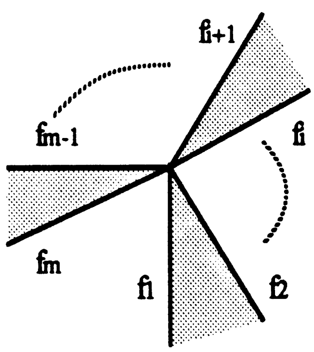


Figure 7. Well-adjacency of faces.

Procedure **MWCS_FACES** (f, Ω_i)

/* Find those faces of a maximum well-connected set of pseudo polyhedron Ω;
f is the index of a face that is required to be on the MWCS.

*/

begin

step 1. output f

step 2. $e_1, e_2, ..., e_k$ <-- edges of face f

step 3. for j=1, k do

step 3.1. f' <-- the index of the face well-adjacent to f at edge e_j

step 3.2. if (f' has not been output) then call **MWCS_FACES**(f',Ω_i)

end do {step 3}

end **MWCS_FACES**

Suppose a total of m edges $e_1,e_2,...,e_m$ of Ω_i are found in a MWCS, denoted by P, through **MWCS_FACES**. Let k_i and k_i' be the face adjacency indices of Ω_i and P at edge e_i respectively (i=1,2,...,m). Step 1 takes constant time and thus the overall time spent at step 1 when **MWCS_FACES** terminates is O(n), where n is the number of faces on P. Since each face of P is processed only once, the overall time required by step 2 is $O(\Sigma(k_i')\ (i=1,2,...,m))$. As for the loop at step 3, note that the indices of the faces of Ω_i adjacent to edge e_j are stored in the order of their spatial angles in an entry of the E_f list of Ω_i. So, only $O(\mathbf{log}k_j)$ time is needed to locate the position of f in that entry, hence, the index f' of the face well-adjacent to face f at an edge e_j. As a result, the overall time taken by step 3.1 is $O(\Sigma(k_i'\mathbf{log}k_i)$ (i=1,2,...,m)). The total time cost of **MWCS_FACES** is therefore $O(n + (\Sigma(k_i'\mathbf{log}k_i)$ (i=1,2,...,m))).

Before presenting the complete algorithm to carry out the MWCS separation, it is necessary to clarify that, given the indices of n faces that form the boundary of a MWCS of Ω_i, only O(n) time is needed to construct the pseudo polyhedron representation $<V,E,F,NORM,E_f>$ of that MWCS, say P_i. To see this, note that all the V,E,F,NORM and E_f lists of P_i are readily available in the $<V,E,F,NORM, E_f>$ of Ω_i. The only work needed besides the retrieval is to re-index the vertices, edges and faces of P_i once they are retrieved from Ω_i. For example, if only vertices $\{v_3,v_4,v_7,v_9,v_{15}\}$ are on P_i, and there is an edge on P_i whose entry in the E list of Ω is <9,4>, then this edge will become <4,2> in the E list of P_i because vertices v_9 and v_4 now sit at the forth and second positions of the V list of P_i. Analogously, if edges $\{e_2,e_7,e_{10},e_{13}\}$ are on P_i and P_i has a face stored in the F list of Ω_i as <10,7,13>, then this face will become <3,2,4> due to the re-indexing of $\{e_2,e_7,e_{10},e_{13}\}$. Clearly, this re-indexing process can be done in O(n) time through simple index mapping. Let **MWCS_OUTPUT** be such a process, which takes as input a pseudo polyhedron Ω_i and a list L of indices of the faces of Ω_i and outputs the pseudo polyhedron representation of a MWCS of Ω_i whose faces are those of Ω_i with indices in L. Utilizing both procedures

MWCS_FACES and **MWCS_OUTPUT**, the algorithm given next performs the MWCS separation.

Algorithm **MWCS_SEPARATION** (Ω_i)

/* Compute the MWCS's of a pseudo polyhedron Ω_i and output them */

begin

step 1. unmark all the faces in the F list of Ω_i

step 2. while (there is a face f in F which is not marked) do

step 2.1. L <-- **MWCS_FACES** (f, Ω_i)

step 2.2. call **MWCS_OUTPUT**(L,Ω_i)

step 2.3. mark all the faces with indices in L

end do {step 2}

end **MWCS_SEPARATION**

Lemma 3. The MWCS separation of a pseudo polyhedron Ω_i with n_f faces can be done in $O(n_f \log n_f)$ time and $O(n_f)$ space.

Proof. In the algorithm **MWCS_SEPARATION**, step 1 takes $O(n_f)$ time. For the **while** loop at step 2, since each face can only be in one MWCS, the overall time cost of step 2.2 and step 2.3 is clearly $O(n_f)$. The time taken by each execution of the procedure **MWCS_FACES** is in the form of $O(n + (\Sigma(k_i' \log k_i)\ (i=1,2,\ldots,m)))$, where n and m are the numbers of the faces and edges on that particular MWCS, k_i and k_i' are numbers of the faces of Ω_i and that MWCS adjacent to an edge of the MWCS respectively. By the same reason that a face of Ω_i can only be in one of its MWCS's, the sum of $\Sigma k_i'(i=1,2,\ldots,m)$ over all the edges of Ω_i is $O(\Sigma k_i\ (i=1,2,\ldots,n_e))$, where n_e is the total number of edges of Ω_i. Therefore, after the termination of **MWCS_SEPARATION** the overall time taken by step 2.1 is $O(n_f + (\Sigma k_i) \log n_f)$, that is, $O(n_f \log n_f)$, since Σk_i (i= 1, 2,...,n_e) is $O(n_f)$. Q.E.D.

With Theorem 1 and Lemma 3, the following is in order.

Theorem 2. Whether a pseudo polyhedron Ω_i is strongly non-convergent or not can be detected in O(n**log**n) time, where n is the number of the faces of Ω_i.

It is worth noting that in the ASV process, the algorithm **MWCS_SEPARATION** not only detectes the strong non-convergence of a deficiency Ω_i, but also constructs the MWCS's of the deficiency Ω_{i+1}. The pseudo polyhedron representation of the MWCS's can then be used for the subsequent convex hull and difference operations, along the corresponding branches after Ω_{i+1}.

3. FAST DETECTION FOR ASV NON-CONVERGENCE

An ASV series is non-convergent if it has a non-convergent deficiency Ω_n. A way to detect the non-convergence of an ASV series is to check the non-convergence of every deficiency in the series. The time required by such a detection sets an upper bound.

Theorem 3. It needs at most $O(n^2 \mathbf{log} n)$ time to decide whether the ASV series of a pseudo polyhedron Ω is convergent or not, where n is the number of vertices of Ω.

Proof. Recall that the difference operation is a vertex elimination process. That is, a non-convergent deficiency Ω_k in ASV(Ω) always has fewer vertices than that of Ω_{k-1}. In the worst case, suppose only one vertex is eliminated after each difference operation. To obtain the deficiency Ω_k through ASV process, k convex hull and difference operations are needed, resulting in an overall time requirement of $\sum O(i\mathbf{log}i)(i=n,n-1,\ldots,n-k)$. Therefore, at most $\sum O(i\mathbf{log}i)(i=n,n-1,\ldots,1) \leq O(n^2\mathbf{log}n)$ time is needed to detect whether ASV(Ω) converges or not. Q.E.D.

In an attempt to improve this upper bound, the local cause of the ASV non-convergence of a pseudo polyhedron Ω is sought. Such a study results in a sufficient condition for the ASV non-convergence, which eventually leads to a linear detection algorithm. In search for this local cause, it is useful to invoke the mechanics of ***regularized intersection*** [9].

Definition 5. The regularized intersection of two pseudo manifolds A and B, denoted by A*B, is a pseudo manifold whose interior is $I(A) \cap I(B)$.

Figure 8 gives two examples of regularized intersection. As shown in Figure 8(a), the regularized intersection A*B is the null set ϕ even though the ordinary set intersection $A \cap B$ yields two faces. The result of the regularized intersection in Figure 8(b) is a non-convergent pseudo polyhedron. A tantalizing finding is revealed by the example of Figure 8(b): if there exists a (non-empty) subset (prior to the reqularized intersection) which is non-convergent, then the ASV series to be expanded is non-convergent. Such an observation is not an coincidence, the basis of which is shown by the following lemma.

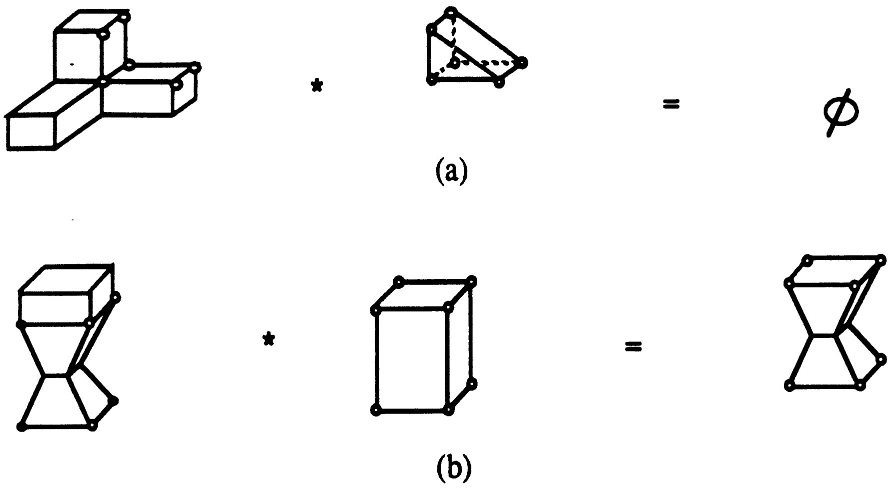


Figure 8. Regularized set intersection

Lemma 4. Let ζ be a subset of the vertices of a pseudo polyhedron Ω. If the regularized intersection between Ω and $CH(\zeta)$ is a non-convergent pseudo polyhedron, the ASV series of Ω is non-convergent.

Proof. Assume that Ω*$CH(\zeta)$ is a non-convergent pseudo polyhedron. It is claimed that all the vertices in ζ are non-eliminatable. Suppose there is a deficiency Ω_i in ASV(Ω), whose vertex set is a superset of ζ, such that some vertex v in ζ is lost on the deficiency Ω_{i+1}. By Lemma 2, this means that all the incident faces of v are the hull faces of Ω_i. Since $CH(\zeta)$ is a subset of $CH(\Omega_i)$, it follows that v is also a weak hull vertex of Ω*$CH(\zeta)$, which is contractory to the assumption that Ω*$CH(\zeta)$ is non-convergent. Q.E.D.

Lemma 4 provides a sufficient condition for the non-convergence of an ASV series, without invoking the ASV process itself. A direct implement of such an algorithm is, however, infeasible since there are O(n!) number of subsets. To reduce this high complexity, the characterization of local subsets of vertices, i.e., those that are adjacent to a common vertex, is investigated.

Let two vertices of a pseudo polyhedron be said to be *adjacent* to each other if they are the two end points of an edge.

Definition 6. A vertex v of a pseudo polyhedron Ω is *supportable* if there exists a plane containing v such that the point set ξ_v lie on its one side, (where ξ_v consists of those vertices that are adjacent to v); otherwise v is a *non-supportable* vertex.

As an example, all the vertices except for v of the pseudo polyhedron in Figure 9(a) are supportable. Also shown in Figure 9(b), a vertex v is non-supportable if and only if it is strictly inside the convex hull of the vertices adjacent to it.

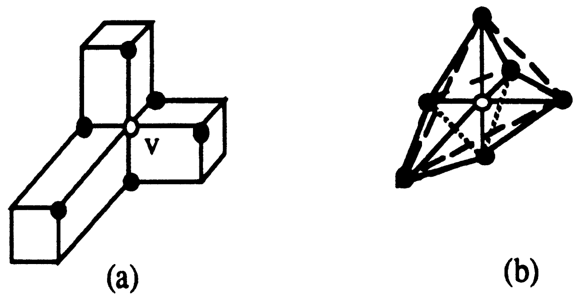


Figure 9. Supportable and non-supportable vertices

Lemma 5. If a pseudo polyhedron Ω has a non-supportable vertex, then the ASV series of Ω is non-convergent.

Proof. Let v be a non-supportable vertex of Ω and ξ_v be the point set consisting of those vertices that are adjacent to v. The lemma is proven by showing that Ω*CH(ξ_v) is a non-convergent pseudo polyhedron.

Since v is internal to CH(ξ_v), all its incident faces have portions that are internal to CH(ξ_v). Then, in each of these faces, there is a point which has an open three-dimensional neighborhood that contains a subset of I(Ω) that is strictly inside CH(ξ_v). By definition of the regularized intersection, this neighborhood is preserved on Ω*CH(ξ_v). In other words, Ω*CH(ξ_v) must be a pseudo polyhedron since its interior is not empty.

Now, consider a hull vertex p of $\Omega * CH(\xi_v)$, as illustrated in Figure 10(a). If p belongs to ξ_v, as none of the incident faces of v can be a hull face of $CH(\xi_v)$, p can only be a strong hull vertex of $\Omega * CH(\xi_v)$. If p does not belong to ξ_v, it must be an intersection point between some face f of Ω and a hull face of $CH(\xi_v)$. (Refer to Figures 10(b) and (c).) Since face f has a portion internal to $CH(\xi_v)$, which becomes an internal face of $\Omega * CH(\xi_v)$, p must be a strong hull vertex of $\Omega * CH(\xi_v)$. Therefore, all the vertices of $\Omega * CH(\xi_v)$ are either internal or strong. By Lemma 2, $\Omega * CH(\xi_v)$ is non-convergent.
Q.E.D.

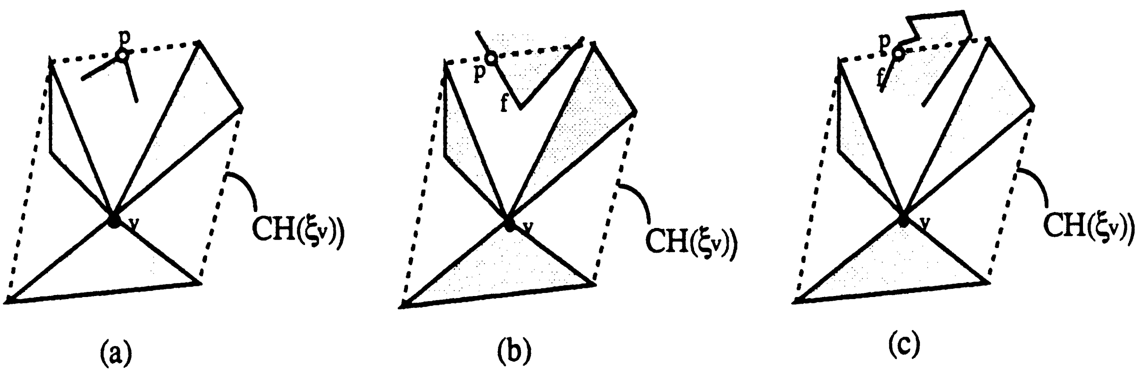
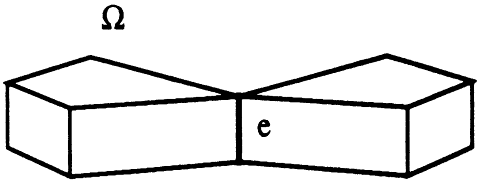


Figure 10. Proof of Lemma 6

As an illustration of Lemma 5, vertex v of the pseudo polyhedron in Figure 9(a) is non-supportable. The regularized intersection between the pseudo polyhedron and $CH(\xi_v)$, where ξ_v are those six vertices adjacent to v, is another pseudo polyhedron as in Figure 9(b) which is non-convergent. By Lemma 5, the ASV series of the pseudo polyhedron in Figure 9(a) does not converge, which can be easily verified.

An extension to the supportability of vertices is the supportability of edges. Consider the pseudo polyhedron Ω in Figure 11(a). Its ASV series can be easily shown to be non-convergent, though all its vertices are supportable.



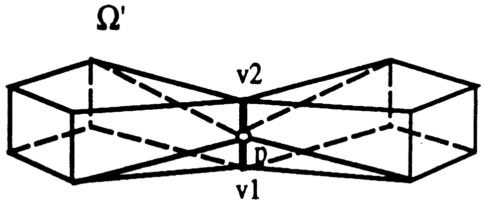


Figure 11. Non-supportable vertex introduced by a non-supportable edge

Definition 7. An edge e of a pseudo polyhedron is *supportable* if there exists a plane containing e such that all the faces incident to e are on one side of that plane; otherwise edge e is non-supportable.

Lemma 6. If pseudo polyhedron Ω has a non-supportable edge e, then ASV(Ω) is non-convergent.

Proof. Assume that the non-supportable edge e has k incident faces $f_1, f_2, \ldots, f_k$, and v_1 and v_2 are its two vertices. Let p be an arbitrary point on e, but not v_1 or v_2. Also let p_i, which is not v_1 or v_2, be a vertex on face f_i $(i=1,2,\ldots,k)$. Since the line segment $[p,p_i]$ is on a face f_i of Ω, the addition of p to the vertex set of Ω as well as the addition of edges $[p,v_1],[p,v_2],\ldots,$ and $[p,p_i]$ $(i=1,2,\ldots,k)$ to the edge set of Ω introduce a new pseudo polyhedron representation of Ω, as in Figure 11(b). Since e is non-supportable, all the points in it, except for possibly v_1 or v_2, are strictly inside $CH(\{v_1,v_2,p_1,p_2,\ldots,p_k\})$. This implies that the vertex p is non-supportable. By Lemma 5, ASV(Ω) is non-convergent. Q.E.D.

It should be mentioned that Lemma 5 and Lemma 6 supply a sufficient but not necessary condition for non-convergence. As an example, all the vertices and edges of the polyhedron in Figure 12 are supportable. Yet, its ASV series is non-convergent.

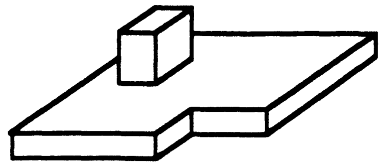


Figure 12. Non-convergent polyhedron with no non-supportable vertices or edges

Nevertheless, a linear time algorithm for detecting the sufficiency of non-convergence offers an attractive alternative to the $O(n^2 \log n)$ time for both necessity and sufficiency.

Let o be the origin and $(x_1,y_1,z_1),(x_2,y_2,z_2),\ldots,(x_k,y_k,z_k)$ be k points in the three-dimensional space. If the point o is supportable against (x_1,y_1,z_1), (x_2,y_2,z_2), ..., (x_k,y_k,z_k), the angle between the normal vector N_o of a supporting plane P_o and the vector (x_i,y_i,z_i) must not be greater than 90° for all the i=1,2,..., k. (See Figure 13.) Conversely, if there exists a vector N_o such that the angle between it and a vector (x_i,y_i,z_i) (i=1,2,...,k) is less than or equal to 90°, then the plane passing through o and orthogonal to N_o is clearly a supporting plane. Therefore, the detection of the supportability becomes the following: Given k vectors (x_1,y_1,z_1), (x_2,y_2,z_2), ..., (x_k,y_k,z_k), find another non-zero vector (A,B,C) such that $Ax_i+By_i+Cz_i \geq 0$ (i=1,2,...,k). It is known [6] that the solution to this three-variable problem, if it exists, can be obtained in O(k) time.

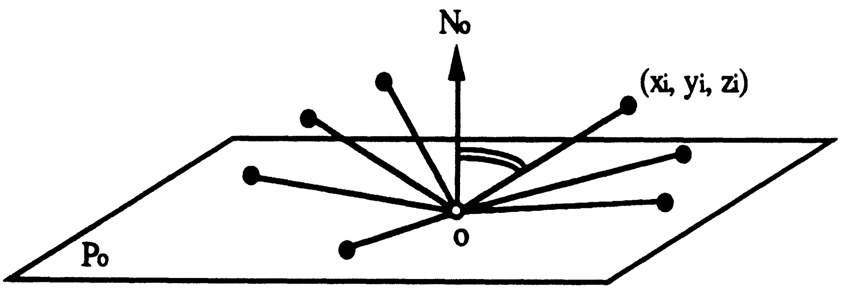


Figure 13. Angular relation between the normal of a supporting plane and the adjacent vertices

Let **SUPPORT**(k,L) be such a supportability detection procedure, which takes a list L of k points as input and outputs either "true" if the origin is supportable against L or "false" otherwise. With the procedure **SUPPORT**, the following algorithm is in order.

```
Algorithm NSV_DETECT (Ω)
/* Detect the existence of non-supportable vertices in a pseudo polyhedron Ω with
   n_v vertices.
*/
begin
step 1.     for i=1,n_v do
step 1.1       v <-- the ith vertex in the vertex list V of Ω
step 1.2.      {p1, p2,...,pk} <-- those vertices in V that are adjecent to vertex v
step 1.3.      translate {p1, p2,...,pk} by a displacement of -v
step 1.4.      if SUPPORT(k,{p1, p2,...,pk})='false' then
                  return with "non-supportable vertex found"
               end if
            end do
step 2.     return with "no non-supportable vertex found"
end NSV_DETECT
```

To device an algorithm for detecting the supportability of an edge e of a pseudo polyhedron Ω, let v and v' be the two end points of e, p be its center point, and $f_1, f_2, \ldots, f_k$ be the faces of Ω incident to e. Also let p_i be a point on face f_i such that the line segment $[p, p_i]$ completely belongs to f_i $(i=1,2,\ldots,k)$. (See Figure 14.) Such a point p_i can be obtained in the constant time from the (clockwise or counter-clockwise) order of the edges. Let **FS**(p, f_i) denote the function which returns the point p_i. Referring to the proof of Lemma 6, e is supportable if and only if p is supportable against the point set $\{v, v', p_1, p_2, \ldots, p_k\}$. This equivalence relation gives rise the following algorithm.

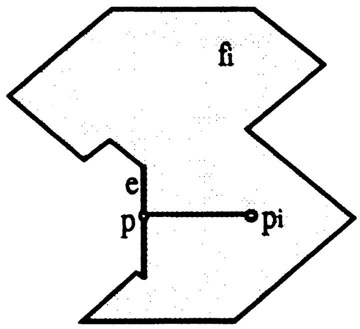


Figure 14. Finding the point p_i on face f_i

Algorithm **NSE_DETECT** (Ω)

/* Detect the existence of non-supportable edges in a pseudo polyhedron Ω with n_e vertices.

*/

begin

step 1. for $i=1,n_e$ do

step 1.1 v, v' <-- the two vertices of the ith edge in the edge list E of Ω

step 1.2. p <-- the center point of [v,v']

step 1.3. $f_1, f_2,\ldots,f_k$ <-- those faces in Ω that are adjecent to the edge [v,v']

step 1.4. $p_1, p_2,\ldots,p_k$ <-- **FS**(p, f_1), **FS**(p, f_2) , ... , **FS**(p, f_k)

step 1.5. translate $\{v,v', p_1, p_2,\ldots,p_k\}$ by a displacement of **-p**

step 1.6. if **SUPPORT**(k+2,$\{v,v',p_1, p_2,\ldots,p_k\}$)='false' then

return with "non-supportable edge found"

end if

end do

step 2. return with "no non-supportable edge found"

end **NSE_DETECT**

Lemma 7. The existence of non-supportable vertices and non-supportable edges of a pseudo polyhedron Ω with n_v vertices, n_e edges, and n_f faces can be detected in at most $O(n_f)$ time.

Proof. The theorem is proven by showing that both the algorithms **NSV_DETECT** and **NSE_DETECT** are $O(n_f)$ in time.

For the algorithm **NSV_DETECT**, because the procedure **SUPPORT** runs in linear time, the time complexity required by the loop at step 1 is linear in $\Sigma d_i(i=1,2,..., n_v)$, where d_i is the degree of the ith vertex in Ω, which has been proven to be $O(n_f)$ in the Appendix of Part I of this paper. Therefore, **NSV_DETECT** runs in $O(n_f)$ time.

For the algorithm **NSE_DETECT**, by a similar reasoning, the time required by the loop at step 1 is linear in $\Sigma k_i(i=1,2,..., n_e)$, where k_i is the face adjacency index of the ith edge in Ω. In the Appendix of Part I of this paper, it is shown that $\Sigma k_i(i=1,2,..., n_e)$ is $O(n_f)$. Therefore, **NSE_DETECT** runs in $O(n_f)$ time. Q.E.D.

4. SUMMARY

It has been established that it takes $O(n^2 \log n)$ time to determine if the ASV seris of a given Ω converges. In particular, it takes $O(n \log n)$ time to detect if a deficiency Ω_i is non-convergent. To remedy the non-convergence, an $O(n \log n)$ time algorithm is offered to separate the culprit deficiency Ω_i into maximally well-connected sets.

As an expedient alternative to the $O(n^2 \log n)$ time detection for non-convergence, the sufficiency condition for non-convergence can be detected in $O(n)$ time.

References

[1] A. Baer, C. Eastman, and M. Henrion, "Geometric Modeling: A Survey" Computer -Aided Design, Vol. 11, No. 5, Sept. 1979, pp. 253-272.

[2] C. Bollobas, Graph Theorey, Springer-Verlag, 1979

[3] A. Cary, "BUILD USERS' GUIDE", University of Cambridge, C.A.D. Group Document,No. 102, Nov. 1979.

[4] B.M. Chazelle, "Convex Decomposition of Polyhedra", ACM Symposium on Theory of Computing, Milwaukee 1981, pp. 70-79.

[5] H. Chiyokura, F. Kimura, "A Method of Representing the Solid Design Process", IEEE Computer Graphics and Application, April 1985, Vol. 5, No.4, pp. 32-41.

[6] F.P. Preparata and M. L. Shamos, Computational Geometry, Springer-Verlag, 1985

[7] F.P. Preparata and S.J. Hong, "Convex Hull of Finite Sets of Points in Two and Three Dimensions", Comm. ACM 2(20), Feb. 1977, pp. 87-93.

[8] A.A.G. Requicha, "Representation for Rigid Solids: Theory, Methods and Systems", ACM Computing Surveys, Vol. 12, No 4, Dec. 1980, pp. 437-464.

[9] A.A.G. Requicha, H.B. Voelcker, "Solid Modeling: Current Status and Research Directions", IEEE Computer Graphics and Application, Oct. 1983, Vol. 3, No.7, pp. 25-37.

[10] A.A.G. Requicha, H.B. Voelcher, "Constructive Solid Geometry", University of Rochester, Production Automation Project, TM-25, Nov. 1977.

[11] K. Weiler, "Topologocal Structures for Geometric Modeling", Ph.D dissertation, Rensselaer Polytechnic Institute, Aug. 1986.

[12] T.C. Woo, "Feature Extraction by Volume Decomposition", Proc. Conf. On CAD/CAM Technology in Mechanical Engineering, Cambridge, Mass. 1982

[13] T.C. Woo, "A Combinatorial Analysis of Boundary Data Structure Schemata", IEEE CG&A, Vol. 5, No. 3, March 1985, pp. 19-27.