Topological Operators for Non-Manifold Modeling

David Marcheix Stefka Gueorguieva

Laboratoire Bordelais de Recherche en Informatique
Université Bordeaux I- LaBRI
351, Cours de la Libération
33405 Talence FRANCE

Abstract

Processing non-manifold topologies (abbreviated as NMT) plays a primary role in current development of solid modeling applications. Euler operators are a powerful tool for creation of valid boundary representations of solids. By contrast with the manifold domain, where topological operators are well understood and implemented, there is a lack of elaboration of their non-manifold counterparts.

In this paper we work out a basis of Euler-like operators for construction, maintenance and manipulation of boundary schemes of non-manifold objects. The presented algorithms are implemented for a NMT data structure and experimental results are discussed in the context of the completeness and soundness of the NMT operators.

Keywords :
geometric modelling, non-manifold topology, boundary representation, Euler operators, data-structures, algorithms.

1 State of the art

A crucial point in current theory and practice of geometric modeling [9] is the representation and manipulation of dimensionally unhomogeneous objects known as non-manifolds.[1] The basic assumption underlying solid modeling, that models are valid i.e. correspond to valid solid objects, implies complex verification tests, especially when non-manifolds are proceeded. The preservation of boundary model validity in the course of NMT object construction and modification is a central issue in the actual study. Two representative results are discussed. Mäntylä's work [8] is concerned with manifold boundary schemes while Weiler's work [12, 11] advocates the non-manifold ones. As it is shown by [8] the topological integrity of boundary representations defined over the the manifold domain can be enforced by simply structural means. For this purpose plane models[2] are used as the mathematical abstraction for the boundary representational schemes. The interest is directed towards the plane models that

[1] A n-manifold is a topological space where every point has a neighborhood topologically equivalent to $D^n(x,r) = \{y \in R^n : \|x-y\| < r\}$.

[2] A plane model is a planar directed graph $\{V, E, F\}$ with a finite number of vertices V, edges E and faces F bounded by edges and vertices. Each face of the graph has a certain orientation around its edges and vertices. Faces, edges and vertices are labeled; if a collection of edges or vertices has the same label, they are identified.

define realizable surfaces.[3] It is proved that a small set of operators can describe all plane models of physical interest. Moreover, the set is sound: no others besides the realizable plane models can be generated. The demonstration is founded on a central result of the algebraic topology, the invariance theorem.[4] The generalization for surfaces with boundary gives the Euler-Poincaré formula[5] and is carried over the manipulation of plane models to identify the set of basic manipulative operations. They can be subdivided into three groups:

. Operations for creation of a skeletal plane model reduced to one vertex, one face model.
. Local topological operations that do not affect the surface genus.
. Global topological operations that alter the surface genus.

Indeed, the simplest plane model of genus = 0, topologically equivalent to a sphere, is the skeletal model. Thus the first operation to begin with in the construction is the skeletal primitive operation. Next, more complex models with given genus can be created through cutting and pasting of faces as long as they do not affect the genus. The cutting joins two vertices of given face with a new edge and thus divides it into two parts. Conversely, the pasting removes an edge separating two faces and thus merges them into a single face. Cutting and pasting operations are interpreted as a pair of inverse operations. By duality, splitting and joining of vertices are also considered in the group of local topological operations. Finally, the connected sum[6] and its inverse operation, the connected minus, permit to generate surfaces with genus greater than zero according to the surface classification theorem.[7]

In summary, all realizable plane models and no others besides can be constructed starting with a skeletal plane model and applying a finite sequence of local and global topological operations.

Boundary representations are the realization of plane models in R^3. A boundary model consists of a geometric part that encodes the embedding of vertices, edges and faces of the corresponding plane model in R^3, and a topological part to save their adjacency relationships. The counterparts of the operations for manipulation of plane models form a complete and sound kernel for the construction and modification of boundary models. The standard notations in use are M and K to denote the make and the kill operations i.e. create and destroy the specified element, and V, E, F, S, H and R, to name the topological entities vertex, edge, face, shell, hole and ring. The kernel operators are:

. skeletal primitive $MVFS\backslash KVFS$.
. $MEF\backslash KEF$ and $MEV\backslash KEV$.
. $KFMRH\backslash MFKRH$.

For convenience, two more operators are introduced $KEMR\backslash MEKR$ which are specially tailored for the manipulation of empty edge loops. The implementation developed by [8] makes use of the Winged-Edge data structure ([1, 2]).

The above operators are often referred as Euler Operators (abbreviated EOP). Using sequences of EOP to process boundary models is useful not only for the preservation of the model validity but also for the flexibility they provide for a multilevel and implementation independent interaction with the underlying data structures. Their practical impact goes beyond the manifold domain. The same approach is applied for the problem of the integrity of non-manifold boundary models.

The extension of the EOP for NMT object processing is made in [11]. The proposed set of boundary graph operators are intended to be used with the Radial Edge structure [12]. Seven basic element types are involved:

The **model** M is the topological modeling space. The **region** R is a volume of space. The **shell** S is an oriented boundary surface of a region. The **face** F is a bounded portion of a shell. The **loop**

[3] If a surface Φ and the boundary $b(A)$ of an r-set A i.e. compact, regular semianalytic subset of R^3, are topologically equivalent, Φ is called realizable.

[4] Let the surface Φ be given as a plane model and let V, E, and F denote the numbers of vertices, edges and faces in the plane model. Then the sum V -E +F is a constant independent of the manner in which Φ is divided up to form the plane model. This constant is called Euler characteristic of the surface.

[5] $V - E + F = 2(S - H) - R$, where S, H and R are respectively the number of connected shells, the genus of the surface and the number of boundary components.

[6] The connected sum of two surfaces is produced if a small disk is cut off from each one and then the surfaces are pasted along the boundary of the cutouts.

[7] Every orientable surface is topologically equivalent either to the sphere, or the connected sum of n tori.

L is a connected boundary of a single face. The **edge** E is a connected set of points forming a space curve which is bounded by a **vertex** V at each end.

Four additional types of element use are associated with face, loop, edge and vertex element types to specify the orientation with respect to the element geometry.

The NMT construction operators are arranged into three categories depending on whether they are specialized to handle a manifold, a non-manifold, or both topologies.

. Manifold operators address solely the manifold constructions. They are denoted with a complementary M and an underscore to distinguish the NMT operators from the traditional EOP names.

MM_EV and MM_E;

. Non-manifold operators.

M_MR, M_SV, M_RSFL, K_V, K_E, K_M, G_V, G_E, G_F,

$ESPLIT$ to split a specified edge in two connected edges,

$ESQUEEZE$ to squeeze the ends of a specified edge and thus deleting an edge and a vertex while adjacency is preserved.

. General operators.

M_EV, M_E, M_F, K_F, U_M_F (element use make face).

In general the NMT operators do not fulfill the Euler-Poincaré equation. The separation of the manifold versions provides a degree of compatibility with existing higher-level functions originally designed with the Euler Operators. Based on the Radial Edge Structure, these operators use the edge as a central mechanism in the manipulations of the boundary. The essential feature is the radial face ordering around an edge. As stated by Weiler, this enables the correct representation of volume adjacency at an edge. Unfortunately, this definition of radial ordering is rather restrictive. Namely, wireframe adjacency at an edge cannot be accurately expressed. Moreover, the radial identification is only carried out at the vertex and edge levels. In practice however, a face adjacency is also possible and may give rise to non-manifold conditions depending on how internal boundary is interpreted. In fact Weiler does not address the problem of the representation and manipulation of the object internal structure. This leads to the necessity of an extension of the radial definition as it is done in section 2.

In addition, it should be remarked that the NMT operators proposed in [11] are not investigated with respect to minimal sets of adjacency relationships. The lack of implementation results makes it difficult to evaluate the practical impact of the NMT operators for the non-manifold manipulation.

The focus of the present article is the study of non-manifold topologies and the computational tools for their construction and manipulation. A set of NMT operators is developed for non-manifolds represented in terms of an elaborated boundary NMT structure, called the radial structure.

The remainder of this paper is organized as follows:

. The radial structure is introduced in section 2.

. Section 3 emphasizes the general presentation of the developed set of NMT operators.

. The extended understanding of the operators along with a thorough description are brought in section 4.

. Section 5 demonstrates on a representative example how NMT objects can be constructed.

2 Radial date structure

Several proposals of non-manifold data structures are known [3, 4, 6, 12]. Unlike the manifold case, no unique mathematical abstraction emerges to underlie the non-manifold boundary models. Promising results are obtained through generalized simplicial decompositions (see [7, 10]). The elaborated radial structure ([5]) is greatly inspired from the complex theory.

The key idea of the proposed scheme is the introduction of the notion of radial link to mark the embedding of different topological elements in the same geometric location that necessitates interpretation of the topological ambiguity. In this way the Weiler's radial edge ordering is relaxed and adjacency at a face is incorporated. Consequently, an uniform way of adjacency encoding is achieved no matter the dimension of the examined topological entities. As a result, a more intuitive object definition is acquired. The object is described as a cellular complex where the n-cells, $n = 0, 1, 2, 3$, can be regrouped into dimensionally homogeneous components with explicit encoding of their adjacency relationships through radial links. Moreover, non-manifold conditions occur at points where the dimensional homogeneity is disturbed and therefore such cases will appear as radial connection between the corresponding valid parts.

Thus two main advantages are achieved:

. Non-manifold conditions at point, edge (loop) or face are encoded uniformly and explicitly while singularities are correctly resolved.

. The internal structure of objects as complex decompositions is preserved while the level of its refinement is under user control.

Formally comparing Weiler's structure with the radial structure, one sees that only one additional element type is introduced, the radial face. However, the new radiality definition and the corresponding cellular decomposition imply a new interpretation of the topological element types.

The notion of n-cell[8] and the notion of complex[9] underlie the element types of the radial structure.

A **vertex** , an **edge** and a **face** are defined as n-cells respectively for $n = 0, 1, 2$.

It should be recalled that cells are not confined to simplices, but that the point set spanned by each cell does not contain non-manifold points.

Next, a **primitive**[10] is defined as dimensionally homogeneous assembly of cells also free of non-manifold deficiency.

Finally, an **object**[11] is a collection of primitives with radially encoded adjacency at vertex[12] , edge or face. It could be a volume, a shell or a wireframe entity, an isolated point or any combination of them.

It should be remarked that the radial connection can be eliminated when the primitives are merged in a kind of manifold gluing. This permits a flexible control on the level of the model refinement.

A **scene** englobes a set of constructed objects and is useful for the structural organization of the entire modeling space.

The **halfelements** are defined similarly to Weiler's element uses. They describe the topological orientation of the boundary surface and enable the interpretation of the non-manifold conditions.

The elements **loop/halfloop/radial loop** are introduced in order to facilitate the manipulation of edge lists as wireframes or boundary contours of faces.

The radial structure gives an appreciable freedom for non-manifold object representation and manipulation as it is illustrated in section 5.

---

[8]A **n-cell** is a connected bounded set such that:

- for each point x of its interior there exists a neighbourhood homeomorphic to $D^n(x,r) = \{y \in R^n : \|x - y\| < r\}$;
- its boundary is divided into a finite number of lower dimensional cells, called the faces of the n-cell;
- a face belongs to the n-cell, if for each point x of the face there exists a neighbourhood homeomorphic to $D^{n+}(x,r) = \{x(x_1, \dots, x_n) \in R^n : \|x\| < r \;\&\; x_n \geq 0\}$.

[9]A **complex** K^d $(0 \leq d \leq 3)$, is a finite set of cells, $K^d = \bigcup\{\sigma : \sigma \text{ is a cell }\}$such that:

- if σ is a cell in K^d, then all faces of σ are elements of K^d;
- if σ and τ are cells in K^d, then $Int(\sigma) \cap Int(\tau) = \oslash$.

The dimension of K^d is the dimension d of its highest dimensional cell.

[10]**primitive** K_p^d $(0 \leq d \leq 3)$ is a connected finite set of d-cells, $K_p^d = \bigcup\{\sigma : \sigma \text{ is a d-cell }\}$ such that:

- if σ is a cell of K_p^d, then the faces of σ that belong to σ are elements of K_p^d;
- if σ and τ are cells in K_p^d, then $Int(\sigma) \cap Int(\tau) = \oslash$;
- for each point x of K_p^d there exists a neighbourhood homeomorphic to either $D^n(x,r)$ or $D^{n+}(x,r)$.

[11]An **object** is a connected coherently oriented complex.

[12]A **radial vertex** defines the set of vertices of an object embedded in the same point of E^3. **Radial edges** and **radial faces** are defined by analogy.

3 Topological operators for non-manifold modeling

The presented set of low level operators enables the construction and modification of non-manifolds in such a way that all intermediate results are valid in terms of the radial structure.

The notations in use are M, K, G, U for make, kill, glue and unglue operations and s, o, p, f, l, e, v, rad for model element scene, object, primitive, face, loop, edge, vertex and radial link for the corresponding element.

The proposed operators can be classified into three categories depending on the element type they proceed.

The first class contains the operators for the building of the simplest non-manifold object that is an isolated point.

$Ms \backslash Ks$ and $Mvpo \backslash Kvpo$

These operators are of the same kind as the Mäntylä's skeletal primitives $MVFS \backslash KVFS$ and Weiler's M_MR and M_SV operators.

At the beginning, the part of interest of the modeling space is initialized as a scene through Ms and then the construction starts with the creation of a single vertex primitive and object made by $Mvpo$. Each operator has an inverse to accomplish the undo operation.

The second category of operators deals with the creation and destruction of the basic type elements.

$Mel \backslash Kel$, $MeKl \backslash KeMl$, $MeKlpo \backslash KeMlpo$, $MeKpo \backslash KeMpo$, $MelKpo \backslash KelMpo$, $Mf \backslash Kf$, $KfMpo$, and $Mvolume \backslash Kvolume$.

The effect of $Mel \backslash Kel$ is very similar to the manifold Euler operators $MEKR \backslash KEMR$. However, no simultaneity exists for the rest of the Me operators as long as they address the non-manifold domain and act on wireframes. For example the creation of a loop in the manifold domain is accompanied with a face creation and therefore $MEF \backslash KEF$ should be performed. Since EOP are intended to construct realizable $2D$ surfaces i.e. boundary surfaces of solids, no equivalence has place also for $Mf \backslash Kf$, $KfMpo$, and $Mvolume \backslash Kvolume$.

A parallel can be established with Weiler's NMT operators. Namely, $Mel \backslash Kel$ can be simulated by $M\_E \backslash K\_E$ and $ESQUEEZE$, $MeKl \backslash KeMl$ correspond to $MM\_E(mekl)$. The rest of Me are partially covered by $M\_E(meks) \backslash K\_E(keflms)$.

There are two important drawbacks in Weiler's description.

Firstly, the effect of some operators is ambiguous. See for example the $M\_F \backslash K\_F$ where $mfl \backslash kfl$ and $mflrs \backslash kflrs$ should be reinterpretated in order to identify whether or not a portion of the region is enclosed.

Secondly, the enumerated NMT operators are not considered with respect to a minimal set. Several operators can be expressed as sequences of more basic ones. See $M\_EV, M\_F(mflrs)$ and $K\_F(kflms)$.

A brief note is made in [11] on how to detect volume closure. In fact this requires a more sophisticated algorithm. The developed $Mvolume \backslash Kvolume$ follow the methodology investigated by [7].

The third class of elaborated operators deals with radial element manipulation. They are specially designed for the creation and destruction of radial links and thus encoding the topological interpretation of non-manifold conditions. It should be noted that some operators belonging to the second group can give rise to radial element modification but only as a secondary effect (see Fig.3 c,d). The $Mrad \backslash Krad$ operators are proper to the radial structure and allow a clear distinction and accurate manipulation of manifold versus non-manifold object items.

As it is shown on the example of section 5, any combination of wireframes, shells and volumes could be constructed in a stepwise manner starting with a skeletal primitive up to the achieving of the final design shape.

Table 1 summarizes the effect of the NMT operators on numbers of topological elements.

	s	*o*	*p*	*f*	*l*	*e*	*v*	*radf*	*radl*	*rade*	*radv*
Ms	+1										
Ks	-1										
Mvpo		+1	+1				+1				
Kvpo		-1	-1				-1				
Mel					+1	+1					
Kel					-1	-1					
MeKl					-1	+1					
KeMl					+1	-1					
MeKlpo		-1	-1		-1	+1					
KeMlpo		+1	+1		+1	-1					
MeKpo		-1	-1			+1					
KeMpo		+1	+1			-1					
MelKpo		-1	-1		+1	+1					
KelMpo		+1	+1		-1	-1					
Mf				+1							
Kf				-1							
KfMpo		+np	+np	-1							
MVolume											
KVolume											
MRadv			[illegible]	[illegible]	[illegible]		+1				+1
KRadv			-1 or 0				-1				-1
MRade			+1 or 0				+2, +1 or 0			+1	+2, +1 or 0
KRade			-1 or 0				-2, -1 or 0			-1	-2, -1 or 0
MRadf			+1 or 0	+2	+2	+n	+n	+1		+n	+n
KRadf			-1 or 0	-2	-2	-n	-n	-1		-n	-n
GRadv		-1 or 0			+1 or 0						+1
URadv		+1 or 0			-1 or 0						-1
GRade		-1 or 0								+1	
URade		+1 or 0								-1	
GRadf		-1 or 0						+1			
URadf		+1 or 0						-1			

Table 1 : *Effect of NMT operators.*

4 Description of the NMT operators

The algorithms for the NMT operators are described along with a specification in the form they are implemented using the C programming language.

The graphic symbols given below are used :

Used for face radial connection	Used for edge radial connection	Used for vertex radial connection	Used for mate edges

Table 2

The destructive operators are detailed when simply inversing of the constructive logic is not sufficient. Output parameters are distinguished from the input ones through a suffix *no*.

Initially, the modeling space is subdivided into scenes, where each one can hold the description of an arbitrary set of objects. In this way several constructions can be managed in parallel and thus improving the design process.

Ms(*Id sno*) **Ks**(*Scene s*)

The *Ms* operator creates an element of type scene (see Fig 1). If not explicitly specified, structure manipulations are performed in the currently opened scene.

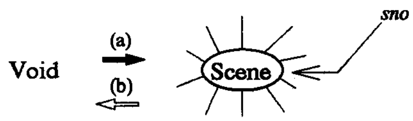


Fig 1 : *(a) Ms, (b) Ks.*

Then the simplest non-manifold can be defined by the means of *Mvpo* operator. A single vertex primitive and object are generated in the given scene (see Fig 2). The desired vertex coordinates are supplied.

Mvpo(*Scene s, Id vno, Id pno, Id ono, Id hvno, float x, float y, float z*)

Kvpo(*Vertex v*)

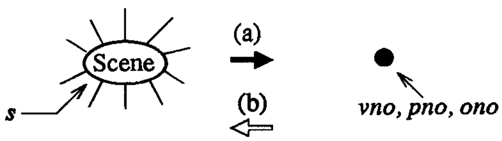


Fig 2 : *(a) Mvpo, (b) Kvpo.*

Next a higher dimensional element (i.e. an edge) can be created with the *Me* operators.

First the instance of an edge that links up vertices from the same object is examined. This causes either the creation (*Mel* on Fig.3) or the destruction of a loop(*MeKl* on Fig.4).

The *Mel* operator creates an edge and closes an edge loop. Different cases are possible depending on:

. whether the edge to be created is a wireframe edge (see Fig 3.a, 3.b) or a shell edge i.e. an edge that is situated on a face (see Fig 3.c, 3.d),

. whether a sole edge loop is generated (see Fig 3.b) or an already existing edge sequence is closed (see Fig 3.a, 3.c).

Mel(*Node n1, Node n2, Id eno, Id lno, Id heno1, Id heno2, Id lno*)

Kel(*Edge e*)

Radial link is generated when the modification provokes a non-manifold condition as illustrated for the shell loop creation (see Fig 3.c, 3.d).

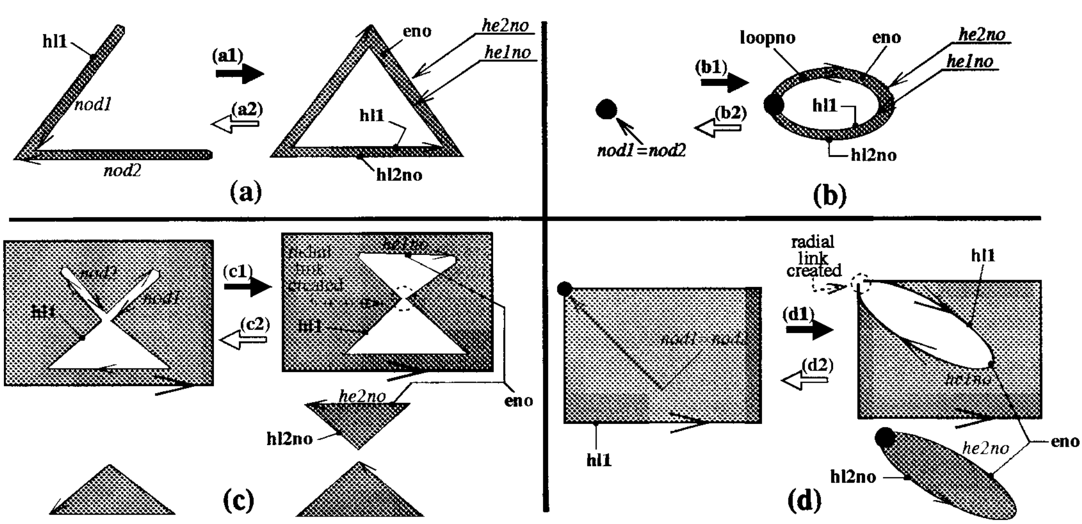


Fig 3 : *(a1,b1,c1,d1) Mel, (a2,b2,c2,d2) Kel.*

The *MeKl* operator is applicable only for shell edge creation when consequently a face loop is destroyed (see Fig 4).

MeKl(*Halfedge he, int what, Node n, Id eno, Id heno1, Id heno2*)

KeMl(*Edge e, Id lno, Id hlno*)

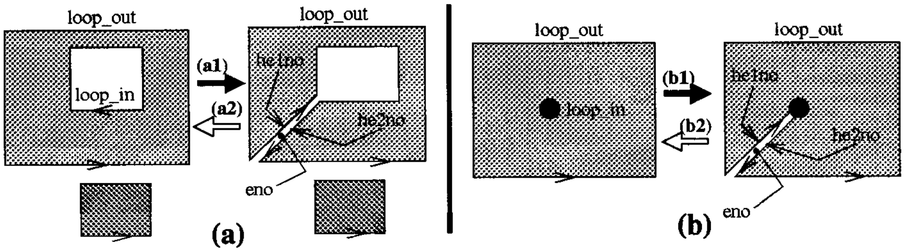


Fig 4 : *(a1,b1) MeKl, (a2,b2) KeMl.*

The next three *Me* operators differ from the ones considered above in that they place an edge between vertices belonging to distinct primitives and objects. For this reason *Me* provokes kill primitive and kill object operations.

For the case when both vertices are from wireframe primitives *MeKlpo* is applied. The kill loop operation is performed on the loop representing the destructed primitive (see Fig 5).

MeKlpo(*Halfedge he1, Halfedge he2, Id eno, Id heno1, Id heno2*)

KeMlpo(*Edge e, Id lno, Id pno, Id ono, Id hlno*)

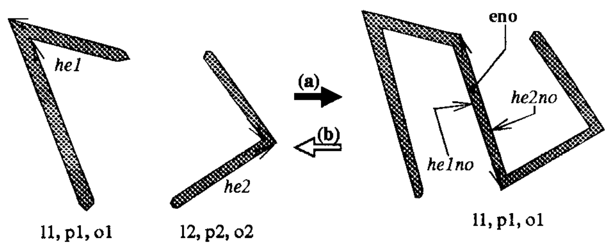


Fig 5 : *(a) MeKlpo, (b) KeMlpo.*

A vertex from wireframe primitive is connected to an isolated vertex primitive through the *MeKpo* operator (see Fig 6).

MeKpo(*Halfedge he, Halfvertex hv, Id eno, Id heno1, Id heno2*)

KeMpo(*Edge e, Id pno, Id ono*)

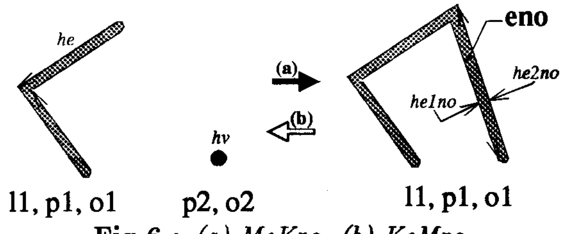


Fig 6 : *(a) MeKpo, (b) KeMpo.*

In the third case of combination of two vertex primitives through *MelKpo* the newly created edge gives rise to a loop generation (see Fig 7).

MelKpo(*Halfvertex hv1, Halfvertex hv2, Id eno, Id lno, Id heno1, Id heno2, Id hlno*)

KelMpo(*Edge e, Id pno, Id ono*)

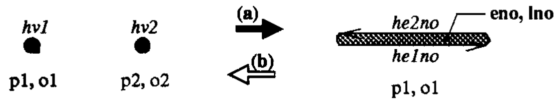


Fig 7 : *(a) MelKpo, (b) KelMpo.*

It should be remarked that if primitives belong to the same object the similar operations could be performed by compound topological operators that first detach the primitives and then call the basis operators.

Further increase of the dimension is performed by Mf operator. That means a wireframe primitive can be transformed into a shell primitive under certain conditions, namely if the wire represents a closed edge loop that becomes the bounding face loop (see Fig 8.a).

Mf(*Loop l, Id fno, Id hfno1, Id hfno2*)

Kf(*Face f*) **KfMpo**(*Face f*)

The inverse Kf operator requires a more complex logic since the face could contain some holes. In that case before digging the face through a Kf operator, the holes should be filled up - i.e. they are separated from the face and put in distinct primitives and objects.

For the Kf operator the following steps are performed:

. Verify if the face belongs to a volume primitive and then apply a *KVolume* to transform the volume into a shell (see Fig 8.b1).

. Test if the face is in a shell that necessitates the isolating of the face from the rest of the shell through a sequence of *Mrade* operators (see Fig 8.b2).

. Transform the face into a wireframe primitive (see Fig 8.b3, 8.a).

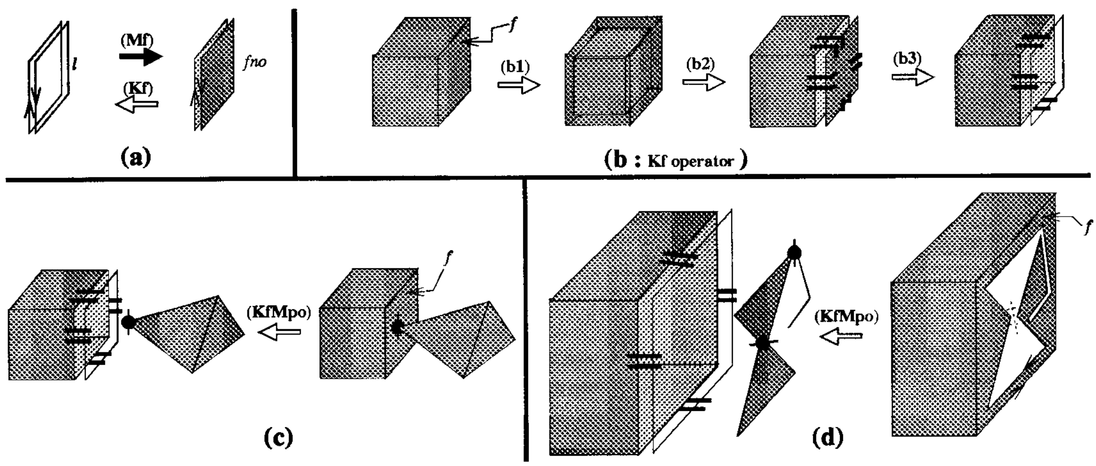


Fig 8 : *(a1) Mf, (a2) Kf, (b) Kf, (c,d) KfMpo.*

The next step in object construction is to form volume components with the *MVolume* operator (see Fig 9). Exploiting the method proposed by [7] candidate shell should verify the requirements for a surface to represent a valid boundary of a solid. An orientation to determine solid interior is given. The same principle is used for the construction of hollows in the object (see the example of the next section).

MVolume(*Halfface hf*)

KVolume(*Primitive p*)

Similarly to the Kf operator, the *KVolume* can provoke the creation of primitives and objects when the boundary of the initial object has a genus greater than zero. For each hole through or cavity a distinct shell is created.

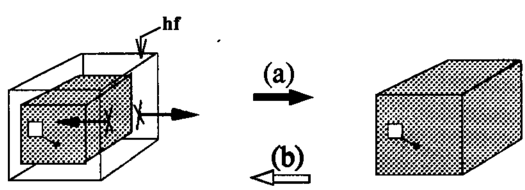


Fig 9 : *(a) MVolume, (b) KVolume.*

Up to here radial elements have been used to encode non-manifold conditions obtained as a side effect of the basic element processing. Next, the operators specially devoted to the manipulation of radial elements are described. The make operators differ from the glue ones in the key feature that they act on a locally manifold part of the objects and unstick it along a specified topological entity. Topological elements are created where the split takes place and radial links are established over the cutout. On the other hand, the glue operators just settle radial links between existing elements. This could also provoke (but as a secondary phenomenon) the generation or elimination of some topological elements (see for example the application of the *GRadv* of two vertices of a closed edge loop that will cause a new loop creation).

If the object connectivity is affected i.e. two separated components result from the cut, make primitive and make object operations are performed to wrap, along with the already existing primitive and object, the newly generated items.

The *MRadv* operator enables the splitting of a wireframe in a specified vertex(see Fig 10.a, 10.b).

MRadv(*Halfvertex hv, Halfvertex hvno, Halfloop hlno, Loop lno*)

KRadv(*Halfvertex hv1, Halfvertex hv2, Halfloop hlno*)

The operator *MRade* enables to tear off a shell along a given edge. It should be noted that the operation is passed on the adjacent vertices (see Fig 10.c).

MRade(*Edge e1, Edge e2, Halfvertex hv1no, Halfvertex hv2no*)

KRade(*Edge e1, Edge e2*)

The operator *MRadf* trims the specified volume through a wedge (see Fig 10.d). As in the case of *MRade* the operation has the consequent repercussions on the adjacent topological environment i.e. creation of faces and their surroundings loops, edges and vertices (see Table 1).

MRadf(*Edge e1, Edge e2*)

KRadf(*Face f1, Face f2*)

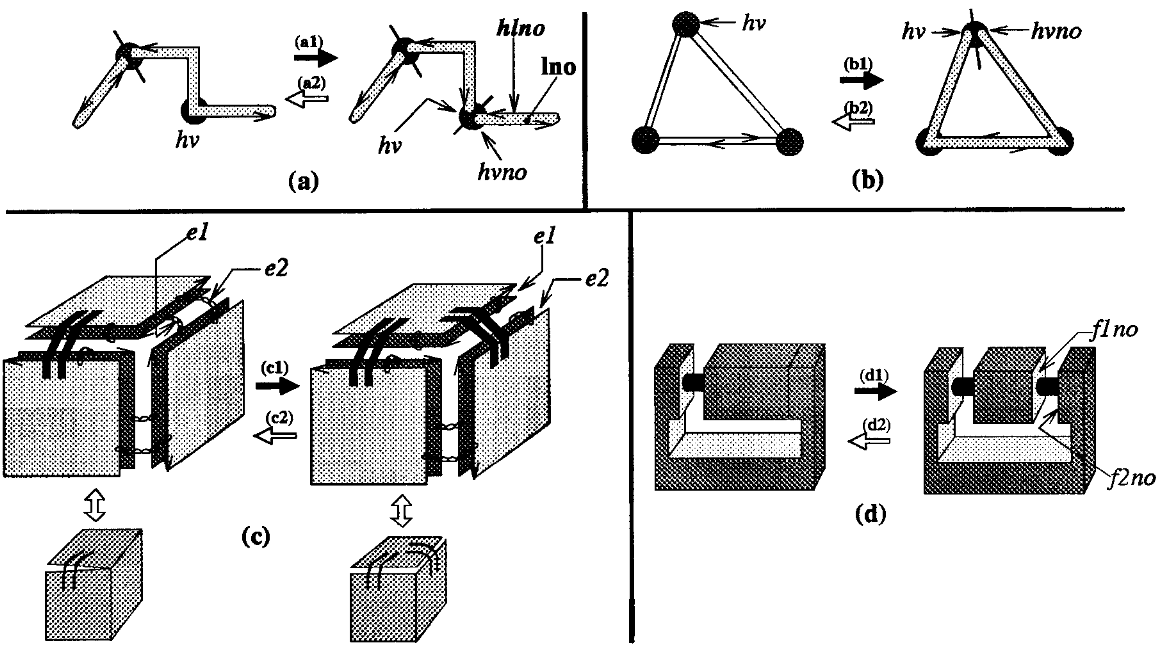


Fig 10 : *(a1,b1) MRadv, (a2,b2) KRadv, (c1) MRade, (c2) KRade, (d1) MRadf, (d2) KRadf.*

The glue operators specified below set up a radial link between two topological elements.

The *GRadv* operator generates a radial vertex. When the radial link is put between two distinct objects just one of them survives (see Fig 11.a).

GRadv(*Vertex v1, Vertex v2*)

URadv(*Halfvertex hv1, Halfvertex hv2, Object ono*)

The *GRade* operator creates a radial edge. It should be remarked that radial connection should already exist on the vertex level (see Fig 11.b).

GRade(*Edge e1, Edge e2*)

URade(*Edge e1, Edge e2, Object ono*)

The *GRadf* operator evolves a radial link between two faces. The surrounding edges and vertices should already been radially connected (see Fig 11.c).

GRadf(*Face f1, Face f2*)

URadf(*Face f1, Face f2, Object ono*)

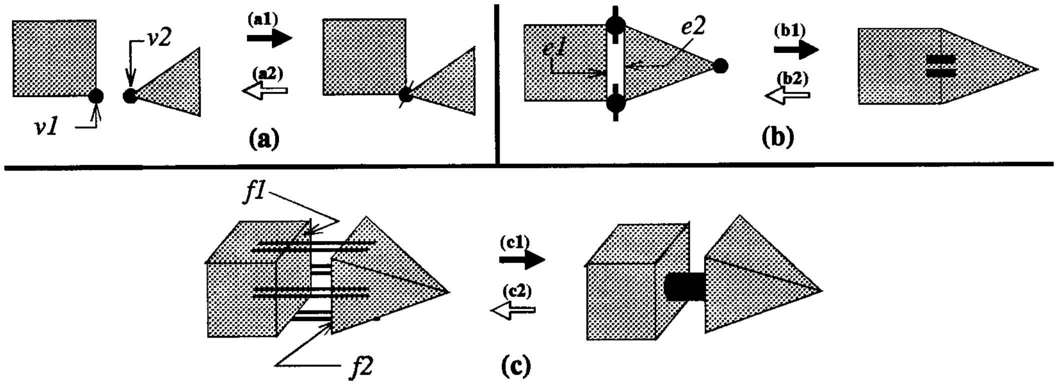


Fig 11 : *(a1) GRadv, (a2) URadv, (b1) GRade, (b2) URade, (c1) GRadf, (c2) URadf.*

The set of operators elaborated above forms the basis for NMT construction and modification. Any object is produced through a finite sequence of the basic operators and thus the model integrity is ensured. A great flexibility of interaction at any level of the radial structure is provided. In some cases however, description becomes cumbersome. This necessitates the development of higher-level operators that enable a more convenient interaction.

For example, in terms of the elaborated operator set the classic *MEV* requires first the creation of an isolated vertex primitive and then an object containing the new vertex. It is then connected with an existing vertex to form the edge. The operation is completed with the destruction of the newly created vertex primitive and object. To avoid such a redundancy, compound operators are developed to perform several basic steps at once. Some illustrations are given in the next section.

5 Example of non-manifold construction

An example of the construction of a non-manifold object is presented. The solid is an assembly of two volume primitives *A* and *B*, a shell primitive *C* and a wireframe primitive *D*.

The modeling sequence is started with *Ms* operator to initialize the portion of space where structure building will take place. Let us detail the construction of the primitive *A*. At the beginning, a *Mvpo* makes a vertex primitive and object. Then a vertex, an edge, and an edge loop consisting of two opposite halfedges are made with *Mevl*. A new edge is added through *Mev* and the wireframe is closed with *Mel*. It should be noted that at this stage there are two halfloops each one corresponding to a given orientation of the wireframe represented as a loop. Next, the loop is transformed into a face boundary loop through *Mf* and thus achieving the description of the first face.

Two adjacent faces share an edge. For that reason the second face is started with a vertex radially connected to an already existing vertex. This is done by *MvRadv*. Next the sequence of *Mevl* and *Mev* defines a wireframe compound by three edges. The last vertex is also glued to a vertex of the existing face with *GRadv*. A *Mel* closes the second wireframe. Loop adjacency is expressed with radial connection performed by *GRade* between the corresponding edges. Then *Mf* solidifies the wireframe loop and completes the new face.

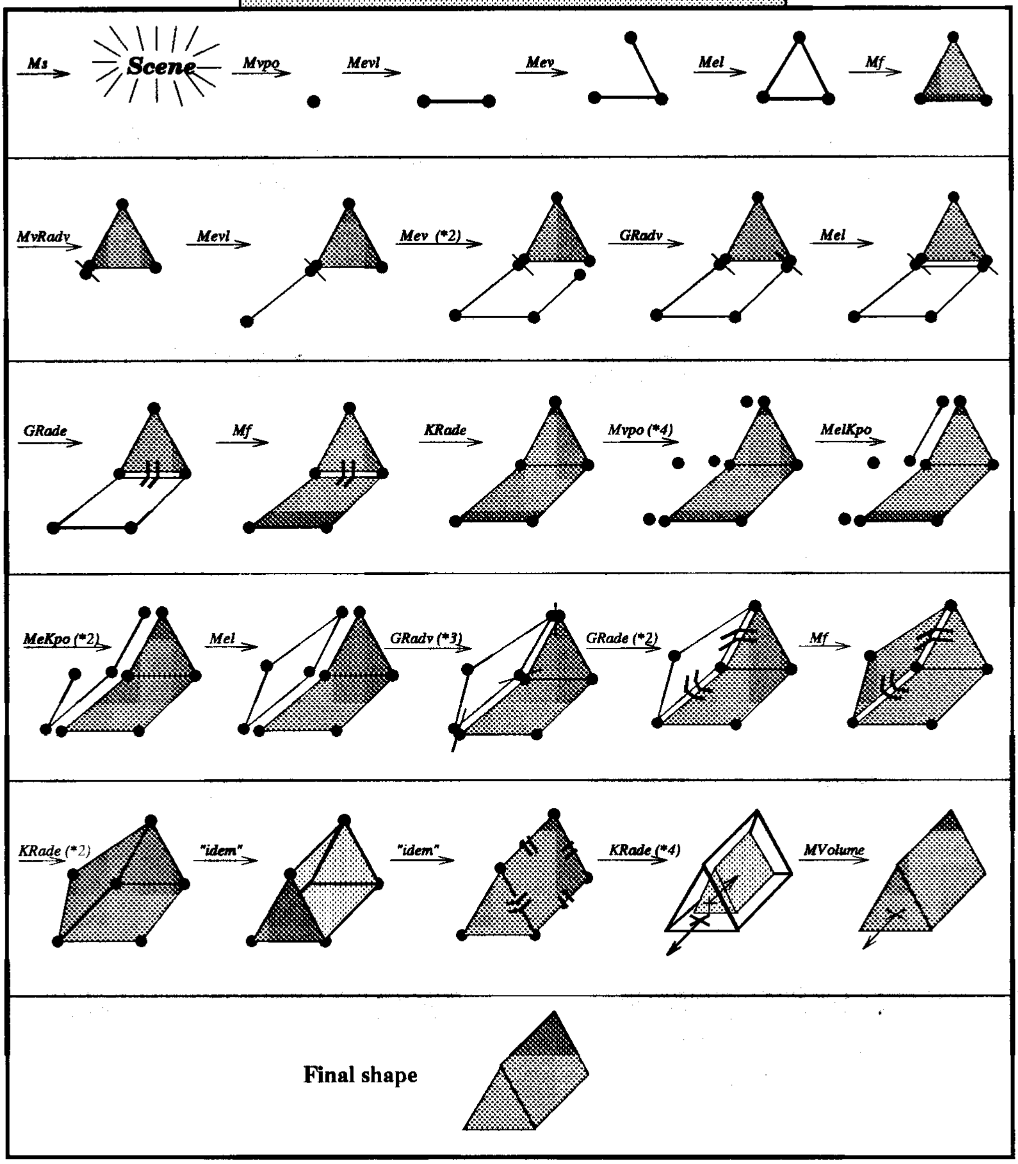
At this step two face primitives are radially connected at an edge. Next, a manifold gluing, *KRade*, unifies them into a single shell.

By analogy, the entire boundary surface can be constructed. The final step is the transformation of the shell primitive into a volume trough a *MVolume* operator. Thus the volume primitive *A* is achieved.

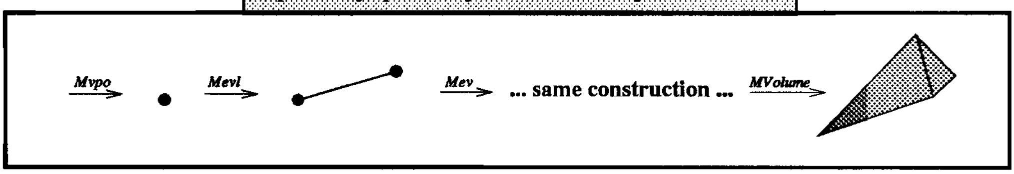
In a similar way one can build the rest of the object as it is illustrated below.

It should be noted that building sequences are not unique and that some compound operators[13] are used for a more convenient interaction. Non-manifold conditions occur at an edge, the contact of A and C, and at vertices, the common vertices of B and C, C and D. Therefore they should be encoded through radial connection between the corresponding topological elements. As for example the triangular pyramid B is radially connected to the face primitive C through a *GRadv* operator.

Sequence of operators for construction of polyhedron A

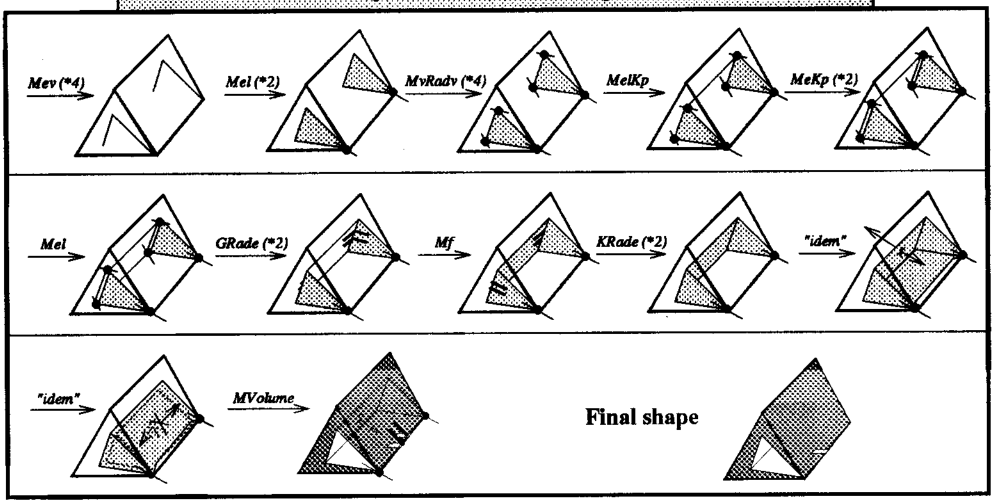


Sequence of operators for construction of tetrahedron B

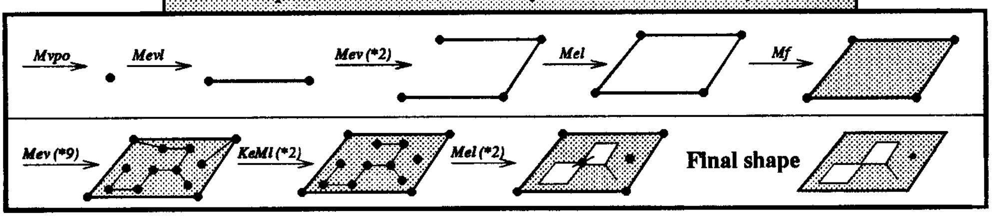


[13]Mev() $\Longleftrightarrow$ *(Mvpo + MeKpo)*
Mevl() $\Longleftrightarrow$ *(Mvpo + MelKpo)*
MvRadv() $\Longleftrightarrow$ *(Mvpo + GRadv)*

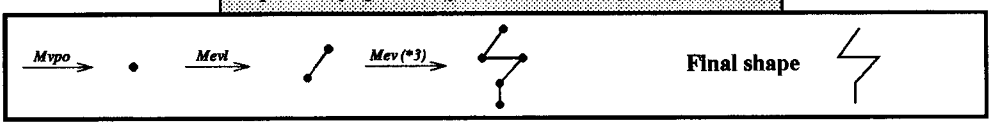
Sequence of operators to dig a hole in the polyhedron A.
The result is a Non Manifold condition in one edge, between A and itself.



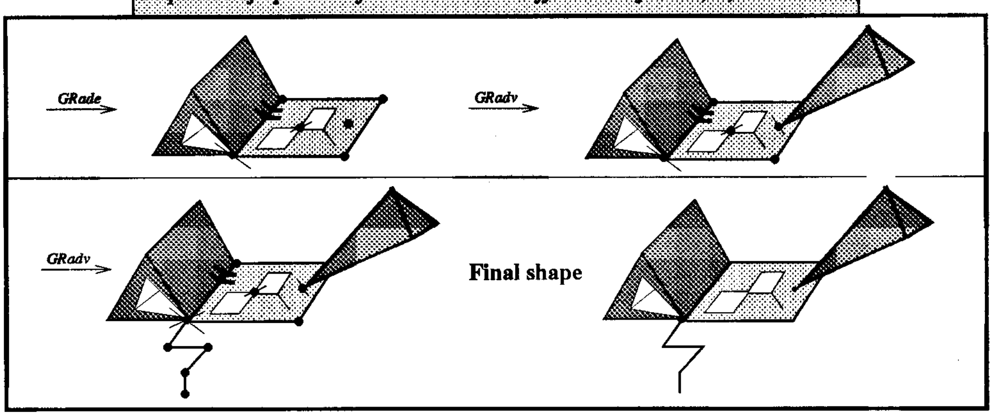
Sequence of operators for construction of dangling face C with holes.
This sequence creates a Non Manifold condition into the face.



Sequence of operators for construction of wireframe D



Sequence of operators for assemble the differents objects A, B, C and D.



6 Conclusion

At this stage of the implementation of the topological operators for construction of non-manifolds, the basic underlying algorithms are implemented on the *C* programming language. They are incorporated in an experimental solid modeller running on Silicon Graphics workstations and that serves as an educational tool. For a better interaction a specialized graphic interface is worked out using *Xmotif* for its flexibility of menu manipulation, and the graphic library *GL* for all realistic visualization utilities.

The goal of the present work is to propose a basis of NMT operators. For the moment the usefulness of the developed operator set is mainly in the elaboration of intuitive constructive techniques for NMT objects that preserve model integrity. An immediate extension is the representation of geometric features as integral parts of the boundary model. Actually, the boolean operations expressed in terms of the elaborated operators are in instance of development. We expect that by the means of NMT operators, geometric intersection giving rise to ambiguous topological interpretations will be correctly resolved.

The further evolution of the advocated modeling techniques naturally brings into consideration other operational issues for the non-manifold processing.

References

[1] B. Baumgart. A polyhedron representation for computer vision. In *Proceedings of the National Computer Conference,AFIPS Conf. Proc.*, 1975.

[2] I.C. Braid, R.C. Hillyard, and I.A. Stroud. Stepwise construction of polyhedra in geometric modeling. In *Mathematical Methods in Computer Graphics and Design*, 1980.

[3] G.A. Crocker and W.F.Reinke. An editable non-manifold boundary representation. *IEEE Computer-Graphics&Applications*, pages 39–51, March 1991.

[4] H. Desaulniers and N.F. Stewart. An extension of manifold boundary representation to the r-sets. *ACM Transactions on Graphics*, 11(1):40–60, January 1992.

[5] S. Gueorguieva and D. Marcheix. Non-manifold boundary representation for solid modeling. In *Proc. of the International Computer Symposium*, 1994.

[6] E.L. Gursoz, Y. Choi, and B. Prinz. Boolean set operations on non-manifold boundary representation objects. *Computer-Aided-Design*, 23(1):33–39, January/February 1991.

[7] Ch.M Hoffmann. *Geometric and Solid Modeling: An Introduction.* Morgan Kaufmann, 1989.

[8] M. Mantyla. *An Introduction to Solid Modeling.* Computer Science Press, 1988.

[9] A.G. Requicha and J.R Rossignac. Solid modelling and beyond. *IEEE ComputerGraphics&Applications*, pages 31 – 44, Septembre 1992.

[10] J.R. Rossignac and A.O. Connor. Sgc: A dimension independant model for pointsets with internal structure and incomplete boundaries. In *Geometric Modeling for Product Engineering*, 1989.

[11] K. Weiler. Boundary graph operators for non-manifold geometric modeling topology. In *Geometric Modeling for CAD Applications*, 1988.

[12] K. Weiler. The radial edge structure: a topological representation for non-manifold geometric boundary modeling. In *Geometric Modeling for CAD Applications*, 1988.