

# **BOOLEAN OPERATIONS ON PSEUDO-SOLID MODELS AND THEIR APPLICATION TO MOLD DESIGN**

Sang Hun Lee<sup>1</sup>, Jin-Woo Jang<sup>1</sup>, Kang-Soo Lee<sup>2</sup>, Sangkun Park, Yoonhwan Woo<sup>1</sup>, Young-Moo Heo<sup>3</sup>, Jin-Seok Yang<sup>3</sup>

Graduate School of Automotive Engineering, Kookmin University, Seoul, Korea<sup>1</sup>  
Division of Mechanical Engineering, Hanbat National University, Daejeon, Korea<sup>2</sup>  
Korea Institute of Industrial Technology, Incheon, Korea<sup>3</sup>

## **ABSTRACT**

This paper describes Boolean operations on pseudo-solid models, and their application to injection mold design. A pseudo-solid is also called a non-watertight solid. It can be defined as a sheet that does not comprise a closed boundary of a solid, due to tolerance or some other problems. When solid models are transferred to a different CAD system, most of them become pseudo-solid models. As most existing commercial mold design systems support only solid modeling capabilities for plastic parts and their molds, the designer has to change pseudo-solids into solids to proceed to the mold design stage. However, this healing work is a very tedious and cumbersome process, and it does not always guarantee the result of a complete solid model. To solve this problem, we proposed and implemented algorithms of Boolean operations on pseudo-solid models. Based on these operations, we developed the splitting capability for core and cavity plates and the modeling capabilities for core inserts and sliders. In addition, we also developed conventional design capabilities for moldbases, ejectors, cooling channels, runners, gates, electrodes, and so on. All the capabilities are modular and can be accessed without following any predefined design sequence. Using these capabilities, the productivity of the mold design can be highly raised by reducing the healing work to a minimum, and the CAM process can be launched early by receiving the surface data of the mold cavity before the healing stage.

## **KEYWORDS**

Boolean Operation; Solid Modeling; Non-Manifold Modeling; Mold Design; CAD

## **1. Introduction**

As solid modeling systems have widely spread in the product design area recently, they are being introduced in mold design for products. To support solid mold design efficiently, dedicated CAD systems for mold design have been developed and introduced to mold manufacturers[1, 2, 3]. Since the CAD systems of mold manufacturers are different from those of customers, CAD data files are delivered in the form of a standard data exchange file like IGES[4]. However, in most cases, complete solid models are not constructed from the product CAD file due to tolerance or other problems. In this paper, such a non-watertight incomplete solid model is called a pseudo-solid

model. The pseudo-solid model can be defined as a sheet body or a group of sheet bodies that cannot form a closed boundary of the solid body. Since the current dedicated systems for mold design have been developed based on solid modeling technology, it is mandatory to heal pseudo-solid models before starting the mold design process. This healing work is a very tedious and cumbersome process that usually takes more than a few days. However, in the view of CAM, a complete solid model is not necessarily needed. Traditionally, surface or sheet models have been used in this area. If the ultimate objective of mold design is to produce a mold, the healing process can be omitted or reduced to a minimum, as most defects of pseudo-solid models are too small to affect the NC tool path generation.

Therefore, in this paper, we propose a mold design system based on pseudo-solid modeling, instead of traditional solid modeling, for improvement of design productivity. To realize this system, we devised the algorithms for the Boolean and sectioning operations on pseudo-solid models, and implemented them using UG/Open API of Unigraphics[5]. The following sections are dedicated to a brief explanation of the algorithm of Boolean operations and its application to mold design. Although the sectioning operation is also used for mold design, we omitted it in this paper, as its algorithm is a variation of the Boolean operation algorithm.

**2. Boolean Operations on Pseudo-Solid Models**

The algorithm for Boolean operations on pseudo-solid models is comprised of five main steps, as described in the flowchart of Fig. 1. This algorithm is devised taking into consideration of the UG/Open API routines. The details of this algorithm are described in the following steps for an operation on two bodies illustrated in Fig. 2.

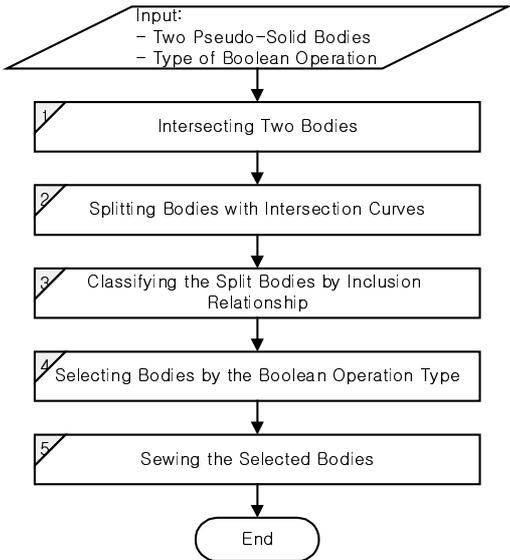


Fig. 1 Overall procedure of the algorithm for Boolean operations on pseudo-solid models

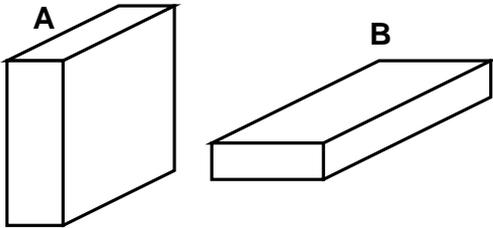


Fig. 2 Target and tool bodies for Boolean operations

Step 1: Intersecting Two Bodies.

The intersection curves between all the faces of a target body A and all the faces of a tool body B are calculated. The faces are classified into intersecting faces and non-intersecting faces.

Step 2: Splitting Bodies.

Each group of the connected non-intersecting faces are extracted and constitute a sheet body. The faces in a group are obtained by traversing the neighboring faces, starting from any seed face in the group without crossing the intersecting faces. Then, each intersecting face is extracted and split by the intersection curves. If the intersection curves on a face are not continuous, they are connected by moving vertices or adding edges. Fig. 3 illustrates sheet bodies extracted from the original pseudo-solids in Fig. 2.

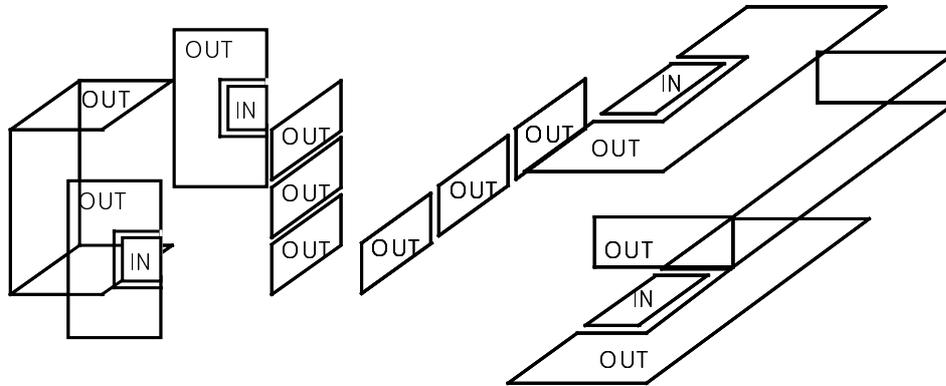


Fig. 3 Sheet bodies obtained by splitting pseudo-solids and their relative locations

Step 3: Classifying the Relative Location of Each Sheet Body.

In this step, the relative location of each of the sheet bodies obtained in Step 2 with respect to its counterpart pseudo-solid is identified. The relative location is represented by IN, OUT, or ON. In case of ON, the location is further classified into ON\_SAME and ON\_OPPOSITE. ON\_SAME means that the normal directions of a sheet and its counterpart pseudo-solid are the same. ON\_OPPOSITE means that the normal directions of a sheet and its counterpart pseudo-solid are opposite.

To determine the relative location, we use a ray test method. In this method, a ray emanated from a point on a sheet in an arbitrary direction is first generated, and then the number of intersections between the ray and the counterpart pseudo-solid are counted. If the number is even, the sheet is classified OUT; if odd, IN. If the ray intersects any vertex or edge, the ray direction is modified until no intersection occurs.

Step 4: Collecting Sheet Bodies.

The proper sheet bodies are collected based upon the type of Boolean operation being applied. Table 1 gives a rule to be followed to collect the proper sheets. For the union operation, the sheets marked by **O** in the column of UNION are collected. For the intersection and difference operations, the sheets marked by **O** in the columns of INTERSECTION and DIFFERENCE are collected, respectively.

		Union	Intersection	Difference
Solid A	OUT	O	X	O
	IN	X	O	X
	ON_SAME	O	O	X
	ON_OPPOSITE	X	X	O
Solid B	OUT	O	X	X
	IN	X	O	O
	ON_SAME	X	X	X
	ON_OPPOSITE	X	X	X

Table 1 Sheet selection rules for each type of Boolean operations

### Step 5: Sewing the Collected Sheets.

The sheet bodies collected in Step 4 are sewn together at their common boundaries. The result can be multiple bodies.

## 3. Application to Mold Design

### 3.1 KMU-MOLD

Fig. 4 shows the design modules of KMU-MOLD, which is an injection mold design system developed by the CAD group of Kookmin University based on Unigraphics. The design modules of KMU-MOLD can be classified into three groups as illustrated in Fig. 4: core design, moldbase design, and miscellaneous support module groups. The core design group includes the modules for creating a pattern from a given product model, generating a core block and its pocket, modeling parting surfaces, splitting the core block by parting surfaces, generating insert cores, and modeling the cavity part of a slider. The moldbase design group includes the modules for generating a moldbase, generating the moving parts of a slider, modeling gates and runners, creating cooling channels, and generating a wide variety of standard parts. The miscellaneous support group includes the modules for providing product and mold information, generating a BOM, generating electrodes for mold cavity production, and so on. The graphic user interface of KMU-MOLD and a completed CAD model for a mold are shown in Fig. 5.

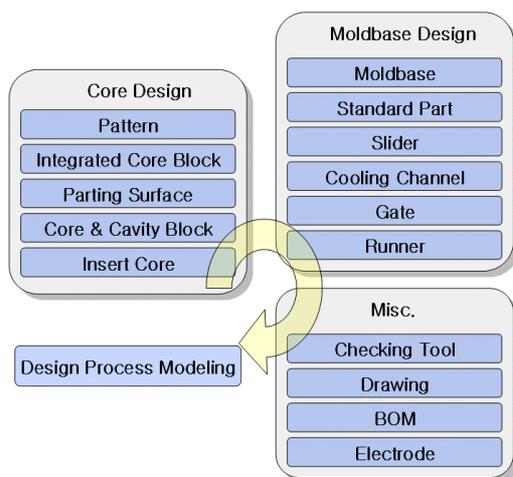


Fig. 4 Design modules of KMU-MOLD

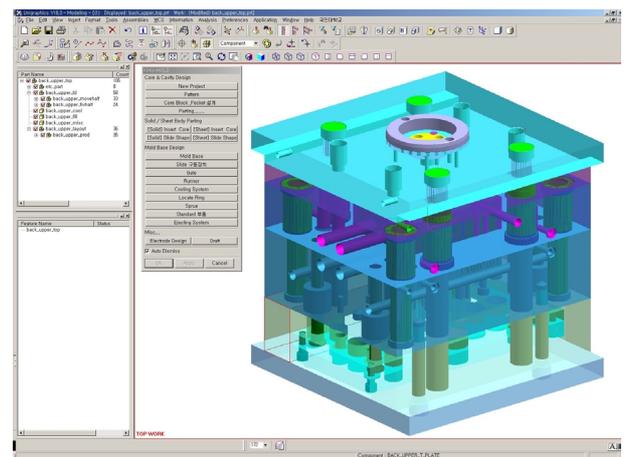


Fig. 5 GUI of KMU-MOLD

### 3.2 Split an Integrated Core Block

Boolean and sectioning operations on pseudo-solids are mainly applied in the modules of core block split, insert core design, slider design, and cooling channel design. Since splitting the core-block is the most important application, we will explain briefly its algorithm below. There are several technical terms in this algorithm. The pattern means a solid or pseudo-solid model that is copied from the customer's product model and enlarged by the shrinkage rate of the specified plastic material to compensate for shrinkage after molding. The integrated core block represents a stock from which the pattern is subtracted for generating a cavity. This integrated block is split along with the parting lines to form the cavity block and the core block. The cavity block is implanted into the cavity plate, and the core block is implanted into the core plate. The shape of the core block is selected from a block, a cylinder, and an extruded solid with user-defined profile.

### Split Algorithm

(Step 1) find the parting lines that are common boundaries of the parting surfaces and the pattern model. The pattern model, the integrated core block, and the parting surfaces for a cover part are shown in Fig. 6.

(Step 2) separate the faces of the pattern into an upper and a lower group by the parting lines.

(Step 3) sew the parting surfaces to the upper face group to get an upper blade as illustrated in Fig. 7(a). Then, using the pseudo-solid sectioning operation, split the integrated core block by the upper blade to get a cavity block. The cavity block is shown in Fig. 8(a). As the product model is a pseudo-solid, the cavity block is also a pseudo-solid.

(Step 4) sew the parting surfaces to the lower face group to get a lower blade, as illustrated in Fig. 7(b). Then, using the pseudo-solid sectioning operation, split the integrated core block by the lower blade to get a core insert block. The core block is shown in Fig. 8(b).

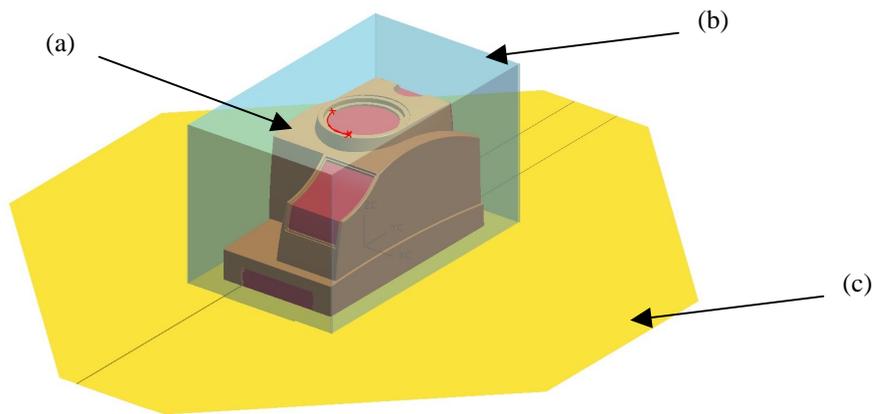


Fig. 6 Examples of (a) a pattern, (b) an integrated core block, and (c) parting surfaces

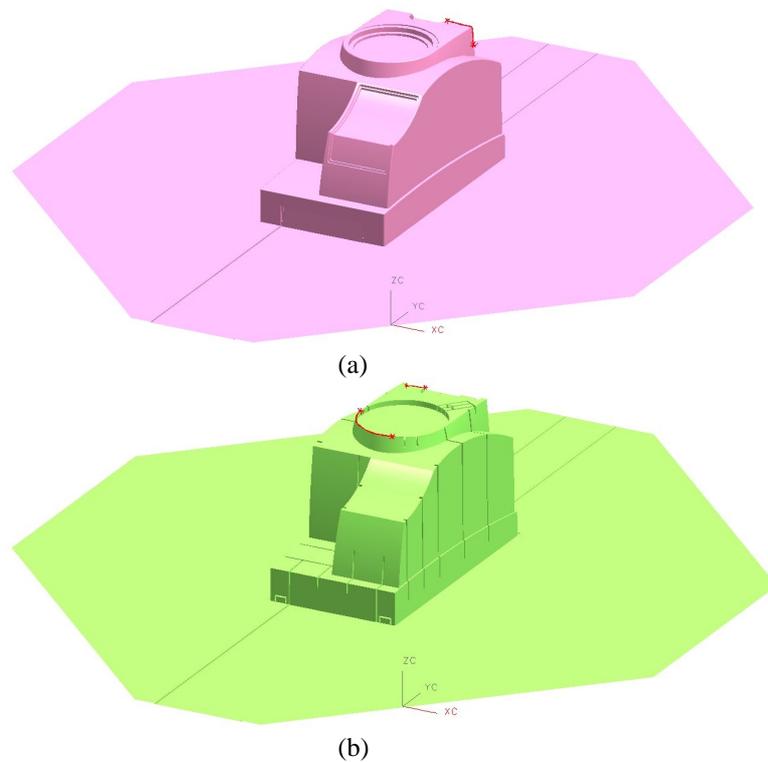


Fig. 7 Upper and lower blades for the split operation: (a) an upper blade, and (b) a lower blade

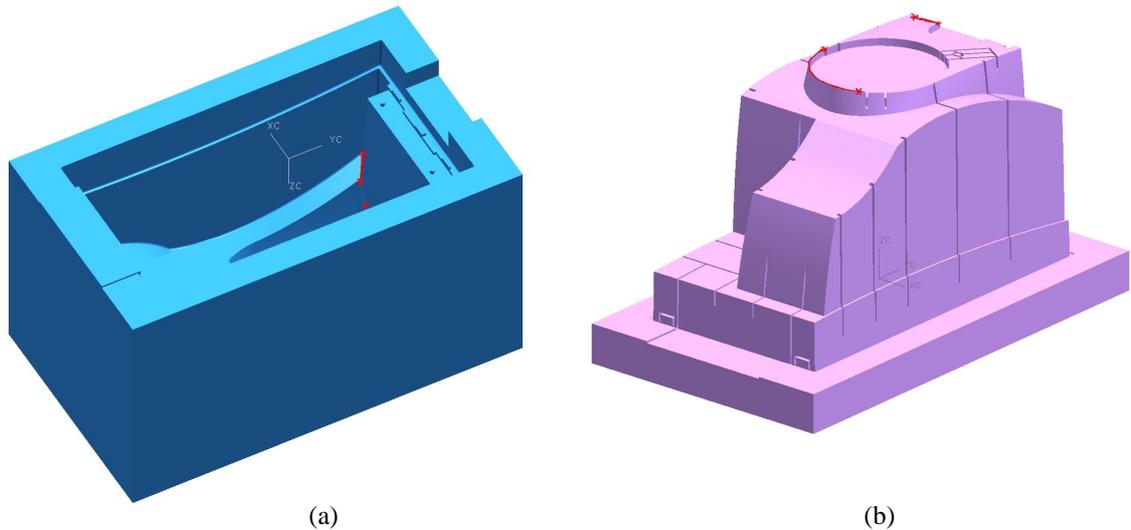


Fig. 8 Split result: (a) a cavity block, and (b) a core block

## 5. Conclusion

In this paper, we proposed an algorithm for Boolean operations on pseudo-solid models, and their application to injection mold design. By introducing these operations to mold design, the productivity of the mold design can be highly raised by reducing healing work to a minimum, and the CAM process can be launched early by receiving the surface data of the mold cavity before the healing stage. However, the current system needs further development for enabling 'wave geometry' capabilities provided by Unigraphics, which automatically modifies the related parts when a precedent feature is modified. In addition, if there is a severe topological defect in the product model, these operations can fail. In this case, the user need to repair the product model to get an acceptable pseudo-solid model.

## 6. Acknowledgements

This work was supported by KITECH.

## REFERENCES

1. Lee, S.H. and Lee, K. (1988). An Integrated CAD System for Mold Design in Injection Molding Process, Production Engineering Division, The Winter Annual Meeting of the ASME, Chicago, PED-Vol.32, 257-271.
2. Fujitsu Ltd. (1997). MOLDWARE CAD user manual.
3. IMOLD (2002), <http://www.eng.nus.edu.sg/imold/>.
4. Lee, K. (1999). Principles of CAD/CAM/CAE Systems, Addison Wesley, Cambridge, U. K..
5. Electronic Data Systems Corporation (2002). Unigraphics Division, UG/OPEN API Reference Version 18.0.