Titre: Title:	A Geometric Approach to Converting CAD Models to CAM Models: an Application on Aeronautical Structure Parts
Auteur: Author:	Hua Lu
Date:	2012
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Lu, H. (2012). A Geometric Approach to Converting CAD Models to CAM Models: an Application on Aeronautical Structure Parts [Thèse de doctorat, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/1062/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/1062/
Directeurs de recherche: Advisors:	Christian Mascle, & Roland Maranzana
Programme: Program:	Génie mécanique

UNIVERSITÉ DE MONTRÉAL

A GEOMETRIC APPROACH TO CONVERTING CAD MODELS TO CAM MODELS: AN APPLICATION ON AERONAUTICAL STRUCTURE PARTS

HUA LU

DÉPARTEMENT DE GÉNIE MÉCANIQUE

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION

DU DIPLÔME DE PHILOSOPHIAE DOCTOR

(GÉNIE MÉCANIQUE)

NOVEMBRE 2012

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

A GEOMETRIC APPROACH TO CONVERTING CAD MODELS TO CAM MODELS: AN APPLICATION ON AERONAUTICAL STRUCTURE PARTS

présentée par: LU Hua

en vue de l'obtention du diplôme de : Philosophiae Doctor

a été dûment acceptée par le jury d'examen constitué de:

M. VADEAN Aurelian, Doct., président

M. MASCLE Christian, Doctorat ès Sciences, membre et directeur de recherche

M. MARANZANA Roland, Doct., membre et codirecteur de recherche

M. BALAZINSKI Marek, Ph.D., membre

M. LAPERRIÈRE Luc, Ph.D., membre

DEDICATION

To my dear wife and my son

ACKNOWLEDGEMENT

The author is greatly appreciative to his supervisors Professor C. Mascle and Professor R. Maranzana, for their encouragement, invaluable discussion and enlightening guidance. I deeply appreciate their efforts for providing me the unique opportunity to pursue my PhD study and bringing me into the CAD/CAM domain. Their wide knowledge and deep insights into scientific problems, sincere attitude and enthusiasm for the research will continue to encourage the author.

Special thanks are given to Dr. O. Msaaf for his constructive discussions and guidance at the beginning of the research. Particular thanks are given to Ms. S. Chalut, Mr. J. Ruby, and Ms. R. Li for their help and encouragement.

Finally, the author would like to express his warmest gratitude to his wife and son for their long-term consistent support and encouragement.

RÉSUMÉ

La conversion d'un modèle de CAO en un modèle de FAO est la première étape de fabrication intégrée par ordinateur. Les principaux problèmes qui concernent la conversion sont les suivants: définir des volumes de matériau amovible géométriquement, vérifier les accessibilités aux volumes ainsi obtenus, associer les opérations d'usinage avec ces volumes individuellement, sélectionner les outils de coupe, mettre en séquençage les opérations d'usinage et assigner une machine pour exécuter le processus.

La détermination des volumes individuels de matériel amovible est le premier problème de la conversion. Dans les dernières décennies, de nombreuses approches ont été développées avec d'énormes efforts, mais aucune étude à ce jour a examiné de manière exhaustive les approches pour générer des volumes de matériau amovible pour traiter des pièces complexes, telles que celles qu’on rencontre dans en aéronautique dans la partie structurelle. Dans la perspective de définir les volumes du matériau amovible, les méthodes existantes se limitent aux fonctions prismatiques. L'objectif principal de cette recherche était de développer des approches systématiques, pour générer automatiquement l'ensemble des volumes de matières amovibles selon les modèles 3D d’une pièce aéronautique structurelle. Il faut alors partir du brut (un morceau de matière première) et usiner toutes les surfaces requises. Grâce à l'outil mathématique disponible des opérations booléennes, il est possible de séparer des géométries volumiques très complexes en volumes plus petits relativement simples. La décomposition du volume delta présente des avantages dans la création des volumes amovibles. Dans cette recherche, les approches de décomposition de volume ont été développées dans le but que chaque volume de matériau puisse être usiné en une seule opération d'usinage. Des arêtes concaves impliquent éventuellement des opérations d'usinage différentes. La détection du bord concave est la première étape de la décomposition de volume intérieur. Dans cette étude, une approche mathématique a été développée afin de vérifier la concavité d'une arête dans la limite d'un modèle solide 3D et une approche de détection des bords concaves est proposée.

Générer des faces de séparation est une étape clé pour définir un volume décomposé. Selon la complexité de l'élément de construction, les algorithmes sont conçus pour créer différents types de décompositions de faces correspondant à des formes locales de la pièce à décomposer. L’union des faces est un outil puissant pour des volumes distincts délimités par des faces de géométries complexes. Cette recherche propose des procédures récursives de décomposition. En utilisant les approches proposées par le modèle de conception 3D, un composant aéronautique de structure est converti en volumes de matériau amovible, nommé sous delta volume (SDV), au moyen de la décomposition du volume delta.

Des approches dédiées à l'identification et la décomposition d’une îlot qui constitue une catégorie spéciale de SDV ont été développées. Nous avons traités les problèmes entourant le cas du trou incliné en détail. Les sujets abordés comprennent la création d'une face d'accès plan, l'expansion du brut, la redéfinition des volumes liés aux SDV et la combinaison de SDV adjacents au trou SDV, etc.

Le deuxième problème de la conversion est de définir les opérations d'usinage correspondant à des volumes individuels. Il est différent de créer des motifs et des zones de cartographie détectés dans une partie de la pièce avec les modèles proposés par d'autres recherches dans les études précédentes. Cette recherche aborde l'attribution des opérations d'ébauche à des SDV qui ont été élaborés en accord avec les propriétés et les positions relatives de la partie face de délimitation spécifique SDV respective. L’affectation d'opérations d'ébauche à des SDV individuels présente des avantages par rapport à une ébauche de toute la partie avec un seul outil. Pour les instances, tout d'abord, pour atteindre une plus grande efficacité de coupe des SDV différents peuvent nécessiter des outils de diamètres différents, d'autre part, très souvent, dans la pratique, il n'est pas nécessaire de balayer un outil dans tout le volume d'un SDV extérieur (un SDV entourant la partie), parce que la séparation du SDV de la pièce peut être réalisée en faisant passer un outil de balayage ou une partie de la surface de délimitation du SDV.

Le troisième problème est de déterminer les caractéristiques d'usinage des SDV individuels. Chaque SDV a ses attributs d'usinage spécifiques, comme la face d'accès, le point d'accès et la direction d'accès, le type et les paramètres de l'outil. Dans cette recherche ces attributs sont déterminés selon les géométries des SDV individuels. Les approches pour déterminer la direction d'accès ont été développées avec l’intention que venant ainsi déterminer l’accès toutes les visages puissent être accessibles, si c'est possible. L'usinabilité d'une partie a été vérifiée pour deux aspects : l’accessibilité angulaire et l’accessibilité linéaire des SDV individuels. Pour vérifier l'accessibilité angulaire, les approches proposées déterminent la posture de l’outil plat et celle de l’espace-outil (notée TPF/TPS). Un TPF/TPS non vide garantit l'accessibilité angulaire d'une

face. Et une intersection non vide de la TPF/TPS de toutes les faces de la part d'un SDV implique qu’il est accessible selon une seule direction. L’accessibilité linéaire a été vérifiée conformément aux orientations spécifiques.

La quatrième question de la conversion dans cette recherche est le séquençage des opérations. Des approches sont proposées pour générer une séquence de mise en service du brut selon les attributs d'usinage et les positions relatives des volumes décomposés. D’autres approches ont été élaborées en vue de renouveler les paramètres de l'outil comme certaines modifications devraient être apportées à la séquence.

ABSTRACT

Conversion of a CAD model to a CAM model is the initial step of computer integrated manufacturing. Main issues concerning the conversion are as follows: defining volumes of removable material geometrically, verifying accessibilities to so obtained volumes, associating machining operations with these volumes individually, selecting cutting tools, sequencing machining operations, and assign a machine to perform the process.

Determination of individual volumes of removable material is the first issue of the conversion. In the past decades many approaches have been developed by enormous efforts but no study up to date has comprehensively discussed approaches to generate volumes of removable material for producing a complex aeronautical structural part. In the perspective of volumetric definition of removable material, existing methods are limited to prismatic features. The main objective of this research was to develop systematic approaches to generating automatically the complete set of volumes of removable material according to the 3D models of both an aeronautical structural part to be produced and the stock (a piece of raw material) to be machined. Due to powerful mathematical tool of Boolean operations available for separating very complex volumetric geometries into relatively simple smaller volumes, delta volume decomposition has advantages in generating removable volumes. In this research volume decomposition approaches were developed for the purpose that every volume of material can be machined in one machining operation. Concave edges imply possible requirement of different machining operations. Detecting concave edge is the premier step of interior volume decomposition. In this study a mathematical approach was developed to verify the concavity of an edge in the boundary of a 3D solid model. Approaches to detecting concave edges were proposed.

Generating splitting faces is the key step to define a decomposed volume. According to the complexity of the structural component, algorithms are developed to create different kinds of splitting faces corresponding to local shapes of the part to perform decompositions. Face union is a powerful tool to separate volumes bounded by faces of complex geometries. This research proposed recursive procedures of decomposition. Using the proposed approaches the 3D design model of an aeronautic structural component is converted into volumes of removable material (named sub delta volume and denoted SDV in this research) by means of delta volume decomposition.

Approaches dedicated to identifying and decomposing an island which is a special category of SDVs are developed. Issues about an inclined hole are discussed in detail. Topics discussed include creation of a planar access face, expansion of the stock, redefinition of related sub delta volumes, and combination of SDVs adjacent to the hole SDV etc..

The second issue of the conversion is to define machining operations corresponding to individual volumes respectively. Different from creating patterns and mapping areas detected in a the part with the patterns as proposed by other researches in previous studies, in this research approaches to assigning roughing operations to SDVs were developed in accordance with the properties and the related positions of part faces bounding specific SDVs respectively. Assigning roughing operations to individual SDVs respectively has advantages over that to roughing the whole part with one tool. For instances, first, to achieve higher cutting efficiency different SDVs may require tools of different diameters, second, very often in practice it is not necessary to sweep a tool all over the volume of an exterior SDV (a SDV surrounding the part) because separating the SDV from the workpiece can be conducted by passing a tool sweeping one or some of the boundary face(s) of the SDV.

The third issue is to determine the machining attributes of individual SDVs. Every SDV has its specific machining attributes, such as access face, access point, and access direction, tool type and parameters. In this research these attributes are determined according to the geometries of individual SDVs respectively. Approaches to determining the access direction were developed under such an intention that from the so determined access direction all part faces are accessible, if it is possible. Machinability of a part was verified from two aspects: angular and linear accessibility of individual SDVs. For verifying angular accessibility, the proposed approaches determine the tool posture flat and tool posture space (denoted TPF/TPS). A non-empty TPF/TPS guarantees angular accessibility of a face. And a non-empty intersection of the TPF/TPS of all part faces of a SDV implies being accessible from one direction. Linear accessibility was verified in accordance to specific direction(s).

The fourth issue of the conversion discussed in this research was operation sequencing. Approaches are proposed to generate an initial operation sequence for roughing according to the machining attributes and relative positions of the decomposed volumes. Approaches were developed to renew tool parameters as some modifications should be made to the sequence.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

$\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$	The position vectors of extreme of CC in directions $\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3$
Brep	Boundary representation
C1, C2	The intersections of PL with F_1 and F_2
CC	Circle
CD	Cell-based decomposition
CE[j]	Concave edge
CV[j], CV[k]	Concave vertex
CV	Concave vertex
DV	Delta volume
DV[0]	The original delta volume
e	Edge
$\mathbf{E}$	The expansion vector
$\{\mathbf{e}_i\}$	Set of edges
$\mathbf{e}_k$	Edge
Echain[j]	Concave edge chain
ec_k	The edge where point *p* is detected
Ex, Ey and Ez	Components of vector **E**
f	Boundary face of SDV[i]
F_1 , F_2	Faces of current DV
fp_0, fb_0	Boundary faces of P , and S
$\{f_j\}$	Set of faces
$\{f_i, f_k\}$	The couple of start and end faces of slot sdm
ff_m	Face filling the gap gp_m
$\{g_j\}$	The updated list of selected faces
gp_m	Gap
g	Boundary face of SDV[j].
$\mathbf{H}_{i0}$	Position matrix of the *i*-th vertex of the original stock

ISO	International Standard Organization
$\mathbf{k}_1,\ \mathbf{k}_2,\ \mathbf{k}_3$	Unit vectors, parallel to **u**, **v**, **w**
$-^*$	The regularized Boolean subtraction
LF	Left faces of an edge
LS	Surfaces of LF
LS[k]	The left surface of an edge
m	The direction vector of the hole axis
$\{\mathbf{n}_j\}$	The set of normal vectors at the centers of faces in $\{f_j\}$
$neg(x)$	Function produces +1 when x < 0, otherwise 0
$\mathbf{n}\mathrm{LF}_k$	The normal of the left face LF_k
nlf	Normal vectors of LF at CV
nRF	Normal vectors of RF at CV
p	The point, on an edge of Echain[k]
P.face	The boundary face set of part P
$\{pieceF[i]\}$	List of face pieces
PL	Plane
$pos(x)$	Function produces +1 when x > 0, otherwise 0
Q_1	The “to” vertex of an edge
Q_2	The “from” vertex of an edge
R	Vector
$\mathbf{r}_k$	The indicator of the half space
RF	Right faces of *e*
$\mathbf{r}_{D0}$ and $\mathbf{r}_C$	Vectors have their end vertex at the original of the reference system
Rs_j	Right surface of the edge $ec_j \in \{ec_i\}$
RS	Surfaces of RF
RS[k]	The right surface of ec_k
S_1 , S_2	Surfaces of F_1 and F_2
SDV.f.open	Set of open faces of SDV

SDV.f.acc	Set of access faces of SDV
SDV	Set or an element of the set of Sub delta volume
SDV[i]	The i-th element of set SDV
SDV[i].bottomF	The bottom face of SDV[i]
SDV[i].open_face	The set of open faces of SDV[i]
SDV.non-part_edge	The set of non-part edges of SDV
SDV.vertex_p	The set of part vertices of a SDV
SDV.vertex_np	The non-part vertex set of a SDV
SDV.geoc_openf	Sets of geometrical centers of open faces of a SDV
SDV.geoc_np-edge	Sets of geometrical centers non-part edges of a SDV
SDV.p_acc	The set of access points to the SDV.
sF	Side face
spltF[i]	The splitting face for sub delta volume SDV[i]
spltSurf[i]	The surface of f[i]
STEP	Standard for the Exchange of Product model data
S.topF	The face of S passing vertex Vtop and parallel to plane XY
$\mathrm{spcN}\left(\mathrm{LS}\left[\mathrm{k}\right]\right)$	The negative half space of surface LS[k]
U	The diagonal matrix with standard unit vectors as its main diagonal
UF	Face union
$(v1, v2)_i$	Vertices of an edge
vd	Diagonal vector
VP0	Set of vertices of initial stock
VP	Set of vertices of the stock after expanding
Vtop	The highest vertex
ε	The expansion matrix associated with a specific expansion vector

LIST OF ANNEXES

INTRODUCTION

Manufacturing is the process of using tools and labor to make things for use or sale. As a general term, it may refer to a vast range of human activity, from handicraft to high tech. But it is most commonly used to refer to industrial production, in which raw materials are transformed into finished goods on a large scale.

From the point of view of economy, manufacturing is a secondary sector of industry, and it is usually directed towards the mass production of products for sale to consumers at a profit. According to some economists, manufacturing is a wealth-producing sector, whereas a service sector tends to be wealth-consuming. The manufacturing industry continuously provides the world with enormous varieties of products for the purpose of daily utilities, improving our life quality, as well as upgrading the means of production. Modern manufacturing includes all intermediate processes required for the production and integration of components that are incorporated into a product.

There are a large number of sectors in manufacturing industry. Among them, aerospace manufacturing, including both aviation and space industries, is one of the most dynamic and attractive sector. As described by the Bureau of Labor Statistics of Canada, aerospace manufacturing is an industry that produces “aircraft, guided missiles, space vehicles, aircraft engines, propulsion units, and related parts”. In 2002, despite the slackening of the world economy, the aerospace industry recorded sales higher than US$250 billion and employed 1,150,000 people. [1]

Canada is a leading player in the global aerospace industry, with industry sales of CAD$21.4 billion in 2003, more than doubling since 1990. The 400 firms making up the Canadian aerospace industry support more than 75,000 employment places for Canadians. Aerospace is a strategic element of Canada’s overall industry. It is the most valuable economic asset and the leading advanced technology export sector of Canada (85% of Canadian aerospace production is destined for export markets). Canada is now the fifth largest exporter of aerospace products in the world.[2]

Montreal and its surrounding areas (Greater Montreal) is the home of Canada’s aerospace industry, the home of third largest aerospace cluster in the world. Greater Montreal is one of the few places in the world where all elements necessary to build an aircraft can be produced within a single metropolitan region. In this area, 170 enterprises providing 35,000 jobs create over 55% of the wealth value of Canada's total aerospace activity.

It is well known that the aerospace industry sector is both capital and technology intensive. To meet the ever increasing requirement for performance and reliability of its products, aerospace manufacturing is equipped with advanced facilities, possessing leading technology for the production of the most complex components of high precision and quality.

Aircraft manufacturing is the greatest proportion of aerospace industry. It is estimated that some 33,000 aircraft at a total value of US$900 billion will be delivered in the next decade. An aircraft is the integration of systems, such as avionics, fuel and propulsion, flight control, communication etc. The airframe provides the infrastructure for the assembly of these systems. Thousands of structural components are mechanically assembled to form a frame that provides the shape of an aircraft and contains its supporting systems.

Airframe structural components are relatively large in size and have low rigidity. These characteristics require special ways of material handling and dedicated fixtures to ensure precision and cutting efficiency. As the number of structural parts is large and their shape is geometrically complex, CAM programming is time consuming using the existing CAD/CAM system.

In aerospace manufacturing, workshops are always under the pressure of increasing their work efficiency, not only because of the large number of components to be machined, but also the frequency of engineering changes made to some of the parts. Design changes are frequent during the customization process of a complex product, and can occur at any stage in the life cycle of a product. A change of one part is not isolated. It usually propagates in the product and causes a series of changes of other parts.

There are many sources and reasons for engineering changes. Eckert distinguished sources of change in two different categories: [3]

- Emergent change, caused by errors or problems in engineering descriptions, found across design stages and throughout the product life cycle;
- Initiated change, arising mainly from an outside source, typically a new requirement from customers or certification bodies, or technical demands from a manufacturer.

Intensive competition is the main factor that pushes aerospace manufacturing companies to shorten the time to market, decrease overall cost of product development, respond to change requests more quickly, and thus keep or win more market share. Effectively integrating their design and manufacturing capabilities is considered one of the most important strategies to enforce an enterprise’s competitive strength. Concurrent engineering and distributed products development, two of the prominent marks of today’s industrial innovation, intensified the challenge of integrating the enterprise’s activities, such as design, manufacturing, and other life-cycle management.

Sophisticated computer aided design (CAD) and computer aided manufacturing (CAM) systems are widely used in the manufacturing industry. Their application has revolutionized the way of creating, saving, and processing product designs and manufacturing plans, evidently increased the productivity of design and manufacturing engineers, and provided essential support for CAD/CAM integration. However, at present, most CAD/CAM integration tools developed to translate product representations between CAD and CAM systems have not been widely adopted by industry. In other words, CAD/CAM research has failed to create any generic way to support CAD/CAM integration, despite the growing need in both traditional enterprises and increasingly global organizations [4].

CAD-CAM integration is the most important cornerstone to concurrent product development, as it provides the possibility for engineering and manufacturing departments to work collaboratively and simultaneously, instead of working in a timely linear way as they traditionally performed. Concurrent engineering does not only mean a gain in time, but also an enormous reduction of product development cost as a lot of changes can be done before many downstream expenses on material, labor, devices, etc. are incurred.

CAPP (Computer aided process planning) is the bridge between CAD and CAM. The main functions of a CAPP system are converting a CAD model to machinable features, associating a machining operation with each feature, selecting a tool with parameters specifically suitable for one of more of the machining operations, selecting machines to perform all or a sub-set of the operations, sequencing machining operations in the order as a piece of raw material is converted into a finished part step- by-step. A complete integrated CAD/CAM system should be able to perform at least the following activities automatically:

1. Dividing the volume of removable material into smaller volumes (in this research defined as sub delta volumes), each of which can be removed by a single specific cutting operation;
2. Associating geometric representations of each sub delta volume with the proper machining operation;
3. Determining the tool parameters and tool approach direction (TAD) for each operation, as well as judging the machinability of the part with a given set of tools and cutting processes;
4. Determining the required number of axes of a machine tool for performing all machining operations on a workpiece using a given setup;
5. Sequencing obtained machining operations with consideration of tool access position for each sub delta volume.

Machining is a class of material-removal processes that involves using a power-driven machine tool, such as a lathe, milling machine or drill, to shape metal and other materials. Milling is the main approach used for removing large amounts of material from a workpiece, producing the most complex components of high precision and quality.

An aerospace structural part is usually made from a block of aluminum alloy by a number of mechanical cutting processes with a CNC milling machine. Roughing and finishing are different kinds of operations. Using roughing machining, most of the volume of material of a workpiece is removed, usually in prismatic geometry. Finishing operations create the surfaces of part as they are designed, and ensure dimensional tolerances are satisfied. Roughing operations are actually performed in 2.5D, while finishing usually requires 3 dimensions or more. In the machining process, a workpiece is held fixed, either by a vise or clamped firmly against a supporting face.

Available commercial CAD/CAM systems use CAPP functions to adapt to certain circumstances, but obvious limitations in milling operations are noticed. Many functions are still done manually. For example; machining area indicating, operation defining and sequencing, tool parameters and TAD determining, reference machining axis changing etc. In current systems, geometries of intermediary models for roughing and finishing usually need to be created by a process programmer. Functions such as analyzing machinability and determining the required number of axis of machine tools are not included in these systems.

This dissertation tries to develop systematic approaches for overcoming the shortcomings of existing CAD/CAM systems listed above. With these approaches, operation sequences can be generated directly from the CAD model of a part and a 3D stock and, at the same time, tool parameters and access points are generated. The question of machinability is answered from the point of view of geometry.

The following sections of this dissertation are organized in the following manner:

Chapter 1 contains a review of published literature on related topics, including machining feature recognition, automation of machining process planning, and milling cutter selection.

Chapter 2 presents the basic concepts and methodology used in this research.

Chapter 3 presents our work on the geometry of removable material, including approaches for decomposing delta volume into sub delta volumes which converts a CAD model of a part into a CAM identifiable model geometrically. Approaches for identifying attributes such as open/close face, top/bottom face are developed.

Chapter 4, based on the achievement of decomposition, describes our work on generating basic manufacturing information, such as access point, tool approach direction, tool type and parameters etc.. Both surface and volumetric information have been taken into account in determining tool parameters to distinguish tools that are efficient for roughing from those used for surfacing. In this chapter, to improve the volume decomposition approaches, subjects related to an inclined hole SDV, such as requirement of stock expansion and redefining adjacent SDVs are discussed.

In this chapter machinability of a sub delta volume is investigated from two aspects: angular and linear accessibility.

Chapter 5 gives an example to illustrate the complete procedures to convert a CAD model of an aeronautical structural part into corresponding CAM model by applying the approaches proposed in this research.

CHAPTER 1 STATE OF THE ART

Efforts to facilitate data flow from design to manufacturing process using numerical control (NC) technologies is one of the main sources that resulted in CAD development. It is this resource that brought about the linkage between CAD and CAM. In recent years, integrating design and manufacturing functions of CAD/CAM-based production processes has been one of the most important trends in CAD/CAM technology development.

Integration of computer aided design (CAD) and computer aided manufacturing (CAM) is considered the first step and primary objective of a computer integrated manufacturing system (CIM). The objective of CAD/CAM integration is to realize automatic transfer and conversion of product data among computer-based product development and manufacturing packages. Thus product information can be further used by downstream applications for the purposes of post processing, machining process simulation and so on.

The total integration of CAD and CAM packages into a common environment is still under development. Many of the major developments have been uncoordinated and a great deal of overlap exists in their intended functions [5]. Traditional computer aided product development/manufacturing application systems, such as CAD, CAM and CAPP (computer aided process planning), have been developed separately; each of them has functions intended to meet the needs of a specific department in an enterprise to perform production activities more efficiently. Although partial benefits were gained on improving performance of branches independently, automatic information transmission and inter-system exchanges were not realized due to lack of global programming [6]. Commercial CAD/CAM systems are powerful for geometric definition, and CAM systems are mostly limited to CNC programming. Computer aided process planning systems play the role of inter-mediator between CAD and CAM systems, but the kind of systems available in the market are incomplete and limited when compared with the number of CAD and CAM systems available [7]. Because the drawing and related technical files exported from CAD systems are not understandable to a CAPP system, a process programmer doing process planning usually has to re-engineer the CAD-defined part to convert it to a proper data format that is recognizable to the CAPP system. From CAPP to CAM, information transfer also needs a great amount of interaction between man and computer. This is one of the main shortcomings in today’s CAD/CAM systems, and it not only decreases work efficiency and increases product development costs, but may also result in errors during data transmission and conversion and reduce reliability of product data.

A great portion of research in CAD/CAM integration focuses on three topics: feature recognition, feature-based CAD/CAM integration and CAD/CAPP integration. There is an enormous amount of literature on this topic, and this chapter provides a brief review of some of these research papers.

1.1 Feature Recognition

Feature recognition is considered as a front-end to CAPP and plays a key role in CAD/CAM integration. It is the process of converting the CAD data of a part into a model that is meaningful and can be manipulated in downstream process engineering and manufacturing activities to produce the part.

A solid model is the core of CAD data of a part. Algorithms for feature recognition typically involve exte——nsive geometric computations and reasoning about the solid model of the part. Since its emergence in the early 1970s , data structures and algorithms of solid modeling have been used in a broad range of applications: CAD, CAM, robotics, computer vision, computer graphics and visualization, virtual reality, etc. [8]. Independent to the solid modeling, computer numerically controlled (CNC) machining intensively stimulates research and development of algorithms for CAM. In spite of wide acceptance and extensive use of CAD/CAM in industry, comprehensive human interaction is still necessary to translate ideas and designs from CAD to CAM in most manufacturing domains [6].

Computer Aided Process Planning (CAPP) is seen as a communication agent between CAD and CAM [4]. An important engineering activity, CAPP determines the appropriate procedure for transforming raw materials into a final product as specified by the engineering design. It is widely accepted that to realize computer integrated manufacturing, CAPP has to interpret the CAD representation of a part in terms of features.

1.1.1 Concept of Feature

Various definitions of features are found in the literature. Mantyla et al. defined features as generic shapes or other characteristics of a part with which engineers can associate knowledge useful for reasoning about the part [9]. In practice, two-level concepts extend the feature’s content. Generic features are low-level features extracted mainly from Brep geometric models. Correspondingly, high-level features are a set of low-level features combined in a user specific manner. A machining feature is defined as a volume of material or a set of part faces that a process planner would consider machining with the same operation [10]. Typically, from the point of view of machining, roughing or finishing a feature is executed with one tool.

Features are application specific. There is a wide spectrum of engineering activities, each of which has its own view of features [11]. For design engineers, a feature might represent functionality; for machining engineers, a feature corresponds to the effect of a cutting operation; for assembly planners, a feature signifies a region of a part which will mate or connect with a corresponding feature of another part; for inspection planners, a feature may stand for a pattern of measurement points. Even within the manufacturing domain, features for casting are different from those for milling [10].

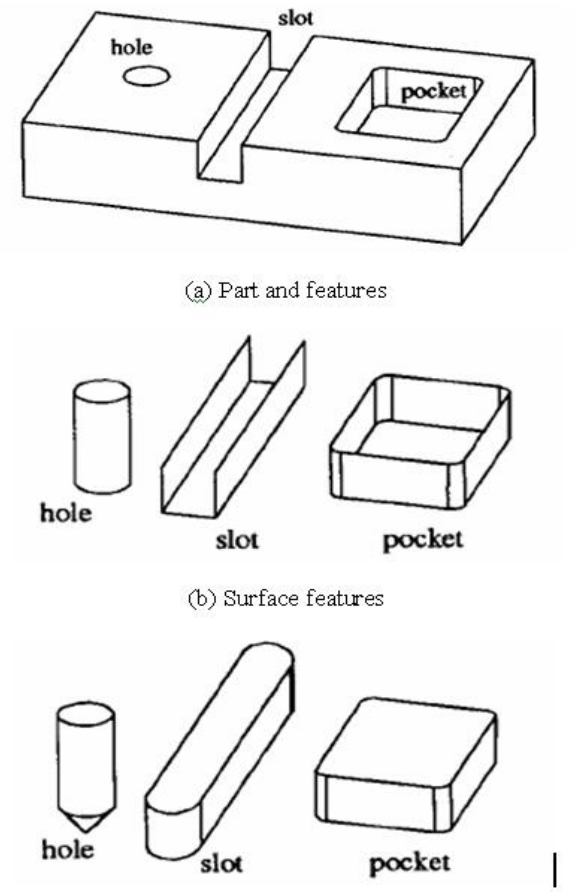
Figure 1-1 shows feature examples: the part is interpreted in terms of a hole, a slot and a pocket. CAPP will use these features to generate manufacturing instructions to produce the part. For example, a drilling operation is usually generated for the hole.

Manufacturing by machining is the application domain that draws most of the attention of feature recognition researchers. Since our research focuses on the machining operation, in this dissertation the terms of manufacturing features and machining features are used interchangeably.

Solid Modeling and Feature Representation

Geometric modeling primitives can be grouped into a number of broad categories (with a good deal of overlap), the most important ones are Brep, solid model, volumetric representation, medial models and IBR (Image-based rendering) models [12]. In manufacturing, three dominant solid representations in use today are; Constructive Solid Geometry (CSG), Boundary Representation (Brep) and Spatial Subdivision [13, 14].

Historically, boundary representation was one of the first computer representations being used for description of polyhedral three-dimensional objects [15]. The reason for Brep becoming the principal solid representation method for most major CAD/CAM systems and also for the input to feature recognition algorithms is that a Brep model defines the entities (faces/edges/vertices) of a solid, so that searching for Brep entity patterns is more promising than searching for CSG patterns, etc.



(a) Part and features

(b) Surface features

(c) Machining features

Figure 1-1 Feature Examples [11]

A machining feature is typically represented in two ways: a surface feature, and a volumetric feature [11], see Figure 1-1. A surface feature is a collection of Brep faces that are to be created by a machining operation. A volumetric feature represents the volume swept by the cutting surfaces of a rotating cutter during machining. Volumetric features provide a more comprehensive representation of actual machining operations than surface features.

1.1.2 Feature Recognition Approaches

Early feature recognition systems were developed by university research groups in the 1980s as they were aware of the need for "post processing" geometry generated by CAD systems in order to drive CAM systems [12]. Many people doubted their value at that time, because productivity gains achieved using 3D CAD in drafting or design were marginal, and these systems were expensive and laborious to use. However, since the task of coding the CNC program for a part could take several days working from 2D drawings, researchers sought approaches to automate this task for significant productivity. Consequently, linking 3D CAD and CAM not only drew academic researchers' attention, but became a commercially important objective for IT business in 3D CAD systems.

Researchers recognized the need for some form of feature or pattern recognition to facilitate the CAD-CAM linkage in the early 1970s. Briad described in his dissertation the first boundary representation modeler mechanical engineering [16].

The term "feature recognition" was first described and named by Kyprianou in 1980 [17]. He established two basic methods for edge classification (vertex based and loop based) and one philosophical concept using generic language (grammar) to express feature definition. His work formed the foundation for many subsequent feature recognition algorithms.

Since these beginnings, voluminous literatures on feature recognition algorithms and concepts have been published. The twenty-five years of development in this domain can be roughly divided into four stages [4].

Stage 1 (1980s to early 1990s): Isolated Feature Recognition. At this stage researchers focused on simple generic features, such as round holes or slots [17-20].

Stage 2 (Early 1990s onward): Interacting Feature Recognition. At this stage researchers invented recognition schemes with the assumption that only traces, or hints, of the geometry might be apparent in the geometric representation of any given part [9, 21].

Stage 3 (Late 1990s Onward): Multiple Feature Interpretations. During this stage, people widely recognized that for a part with interacting features, it was better to submit several alternative feature interpretations to the CAPP system to enable it to find an optimal machining operation schedule. [22] [23] [24]

Stage 4 (2000 to Present): Features on Complex Surfaces. Approaches for recognizing features that involve complex spline geometry were developed [25-27]. Marchetta and Forradellas noticed the need for custom feature representation and developed a hybrid procedural and knowledge-based approach applicable to both classic feature interpretation and feature representation problems. [28]

Han classified the most currently-used feature recognition approaches into three main groups: graph-based, volumetric decomposition and hint-based approaches [11]. In 2004, Di Stefano proposed a new approach for semantic recognition [29]. The principles of these approaches are described in the following sections separately.

1.1.2.1 Graph-based approach

In graph-based approaches, an entire part is represented in a graph data structure, and feature recognition becomes a sub-graph isomorphism problem. Basically, features are associated with graph grammars, and feature recognition processes are accomplished by parsing.

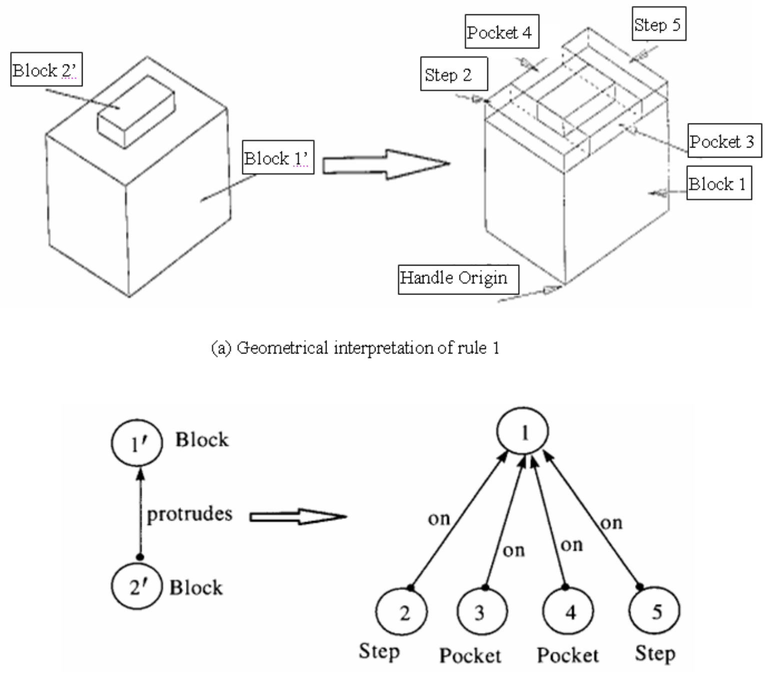
Chuang and Henderson [30] explored graph-based pattern matching techniques to classify feature patterns based on geometric and topological information from the part. In a later work [31], they were the first to note the need to explicitly address both computational complexity and decidability when defining the feature recognition problem. This paper formalized the problem of recognizing features, including compound features, by parsing a graph-based representation of a part using web grammar.

Finger et al. [19] employed graph grammars for finding features in models of injection molded parts. They defined objects to be elements in a language generated by an augmented topology graph grammar and then used their graph grammar to parse a representation of an object. In their approach the design feature model of an object is converted into a graph representation, which is then parsed with a set of application-specific feature graphs to get an application-specific feature model. The advantage of their graphical structure is that it contains both geometric and topological information. Similar work was done by Coles, Kraker and their colleagues [32].

Flasinski proposed a formalism of solid representation on the basis of the parsed family of IE-graphs (indexed edge-unambiguous graph) [33]. The edNLC-type graph grammars were used for the dynamic building and manipulation of such representations [34, 35]. The author believed that with multi-aspect taxonomies defined for features, unique and unambiguous descriptions could be assigned to a part. Flasinski’s work is limited to linear swept features.

Ben-Arieh developed a method based on graph grammar parsing that converts design geometrical features into manufacturing geometrical features for process planning activities [34, 35]. This method extends graph-based feature recognition by supporting embedding transformations, attribute transfer algorithms, decomposition of overlapping features and removal of empty volumes. In the system twenty rules were implemented for prismatic components that can be produced on a 3-axis vertical milling machine. The graph parsing approach treats features as solid models without complicated geometrical analysis. Figure 1-2 shows an example of one of the converting rules. One of the major drawbacks of graph-based feature recognition approaches is the difficulty in identifying interacting features. This is due to the fact that face and edge relationships are changed by feature interactions, and it seems impractical to define all the resulting patterns [14]. Moreover, useful graph grammars do not seem to have efficient parsing techniques. Sub-graph isomorphism is NP-complete problem, the most difficult problem in NP ("non-deterministic polynomial time"), and various heuristic algorithms suggested in the literature cannot break such a complex barrier.

The Hint-based approach was introduced to overcome this kind of pitfall [36].



(a) Geometrical interpretation of rule 1

(b) Graphical representation of rule 1

Figure 1-2 Feature conversion with graph grammar [35]

1.1.2.2 Hint-based Approach

The hint-based reasoning starts from a minimal indispensable portion of a feature’s boundary which should be present in the part, and performs extensive geometric reasoning.

The basic idea of the hint-based approaches is to find traces left by the motion of a milling cutter at the boundary of the part, even when features may intersect with each other [14]. This provides a hint for the potential existence of a feature. These traces are then used to generate a feature volume using geometric completion algorithms [37].

Vandenbrande and Requicha [14] defined the presence rule, which asserts that a feature and its associated machining operation should leave a trace in the part boundary even when other features intersect with it. Furthermore, the presence rule defines the minimal indispensable portion of a feature’s boundary that has to be present in the part [38].

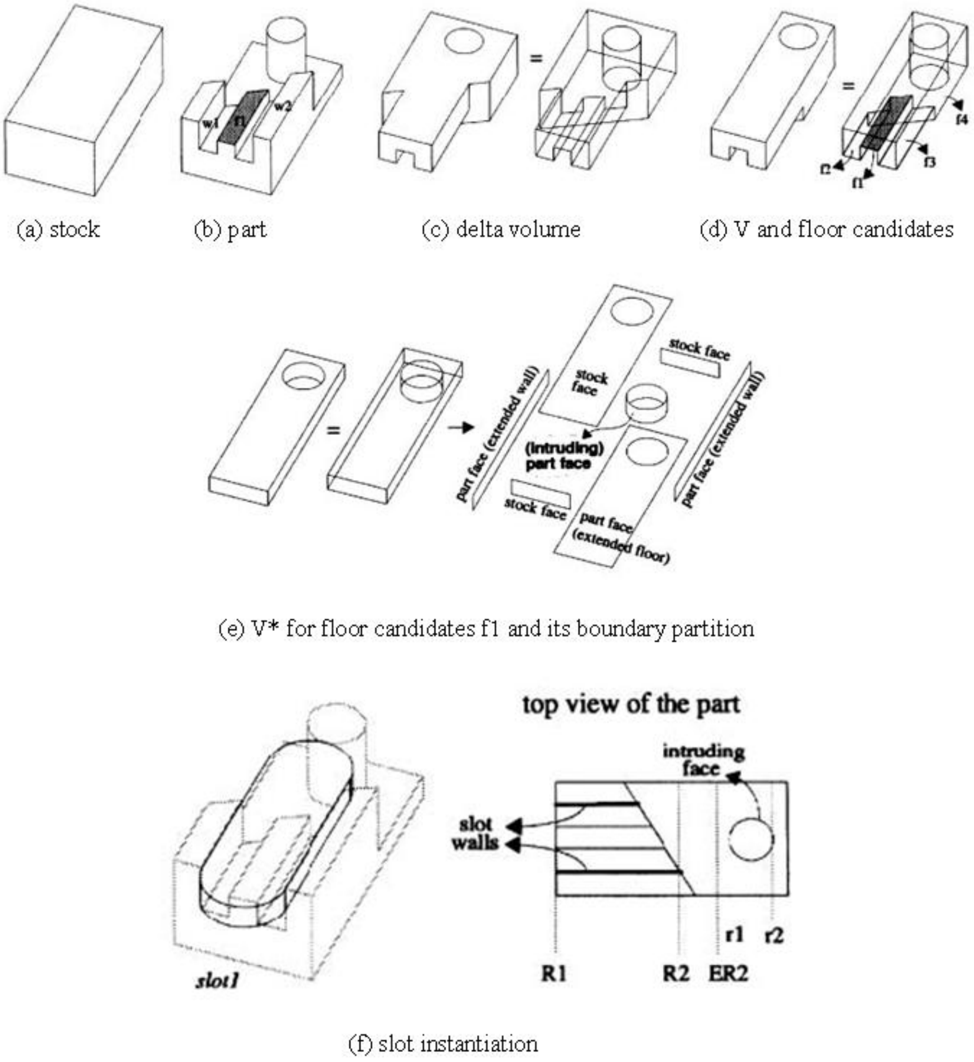
Han’s hint-based reasoning algorithms [39] consist of four main steps: 1) delta volume generating, 2) hint identifying, 3) maximally extended volume V* generating, and 4) testing and repairing. Figure 1-3 shows the step-by-step process for slot recognition.

Hint-based approaches are more successful in recognizing interacting features than the other existing approaches, but they also have some shortcomings [40], because it is quite possible to find traces which are not promising to find a feature, and it is also difficult to find suitable traces for some complex features. Moreover, the existing geometric completion algorithms should be further developed to create more complex pocket volumes.

1.1.2.3 Volume Decomposition

The volumetric decomposition approach decomposes the input object into a set of intermediate volumes and then manipulates the volumes to produce features. Volumetric decomposition approaches are mainly divided into two sub-groups: convex-hull decomposition and cell-based decomposition.

Convex-hull decomposition consists of two steps [22, 23, 25, 37, 41-43]. The first step is to compute the convex hull of a non-convex part, and then to obtain so-called convex deficiency by subtracting the part model from the convex hull. Thus a non-convex part is represented by the set difference between the convex hull and the deficiency see Figure 1-4. This decomposition step is recursively applied to the successive deficiencies until empty deficiency is obtained. Then, the original part is represented by a Boolean combination of convex components. The decomposition step is recursively applied to successive deficiencies, the volumetric contribution of the convex components alternates between positive and negative nature. It is therefore called the Alternating Sum of Volumes (ASV).



(a) stock (b) part (c) delta volume (d) V and floor candidates

(e) V* for floor candidates f1 and its boundary partition

(f) slot instantiation

Figure 1-3 Hint-based slot recognition process [14, 39]

As a result, convex hull algorithms obtain either a positive volume-based or a negative volume-based decomposition of the part. A positive volume-based decomposition represents a part by quasi-disjoint union of element volumes, while a negative-based decomposition describes a part by the set difference between the convex hull and the union of negative components. The latter could be useful for representing the machining removal volumes, i.e. machining features. Figure

1-4 shows the process of convex hull decomposition.

The advantage of these approaches is their applicability to a wide range of manufacturing processes. However, their lack of capability to handle curved features is still one of their drawbacks. These approaches are best at dealing with 2.5D features [43]. It is also doubted whether extraction of form features really does significantly shorten the route to obtain process specific features.

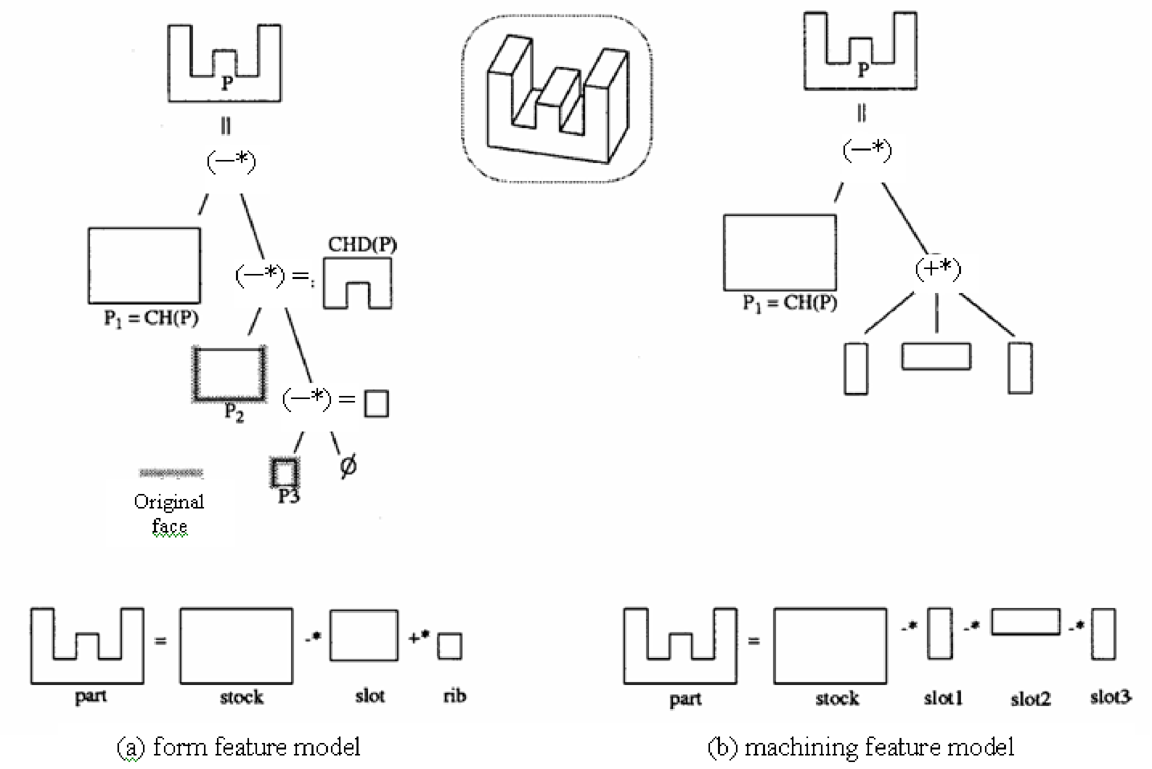
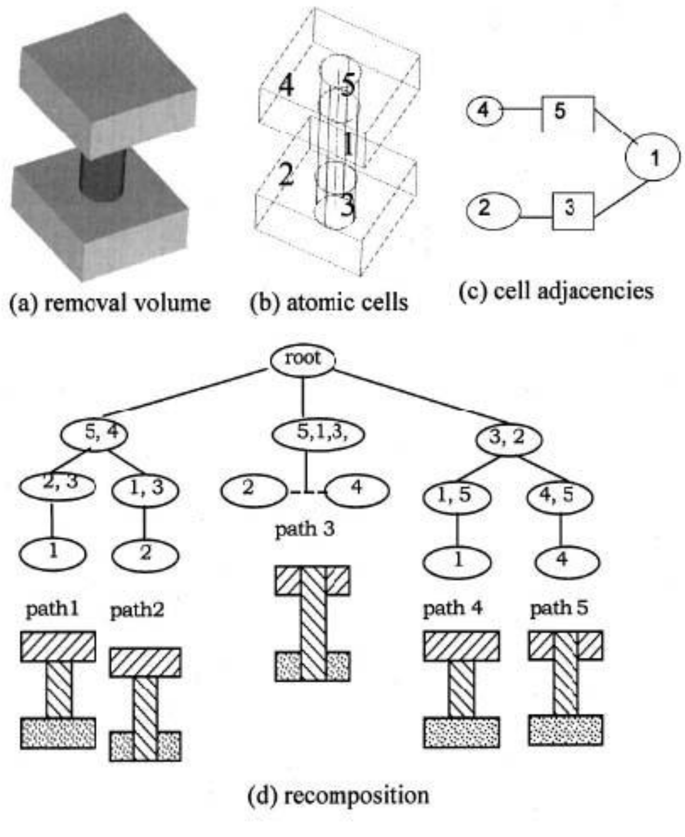


Figure 1-4 Convex Hull Decomposition [23]

Cell-based decomposition (CD) [23, 44] method consists of two steps: decomposition and re-composition. First, an object is subdivided into primitive components that are either disjoint or meet precisely at a common face or edge. Second, these primitive components are put back together to form the object as if they can be glued to each other. Figure 1-5 shows the CD representation for a simple sample. Since engineering objects may be decomposed into constituting components in different ways, in general, an object may have several different CD representations.

For manufacturing feature recognition purposes, the delta volume is taken as an object and the decomposition-recomposition operation is applied to the delta volume. According to Han [11], Sakurai has been a leading advocate for the revival of this type of technique [45-47]. Given a part shown in Figure 1-6-(a) and its stock, the delta volume is decomposed into the cells shown in Figure 1-6-(b). The union of all cells is equal to the delta volume, and the regularized intersection of any pair of cells is null. In the second step, a subset of the cells is combined (composed) to generate a volume to be removed by a machining operation, and in the last step the volume is classified as a machining feature. In the cell-based decomposition approach, the differences of the proposed algorithms mostly lie in the methods for combining cells into features.



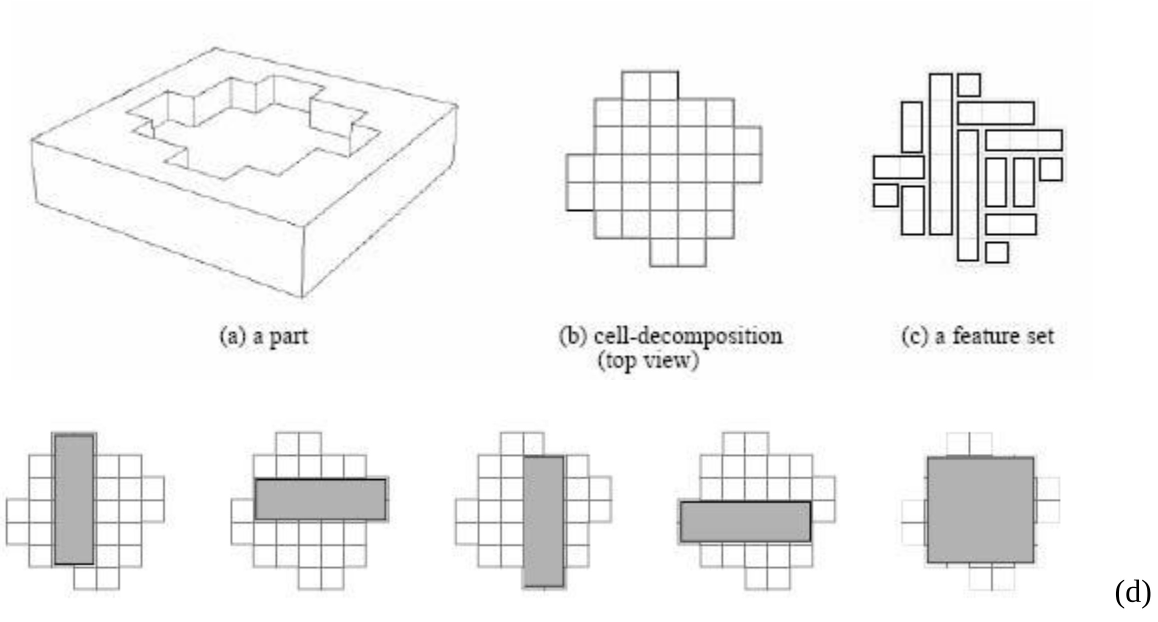
(a) removal volume (b) atomic cells (c) cell adjacencies

(d) recomposition

Figure 1-5 Decomposition and recomposition procedures [11, 48]

Sakurai and Dave (Sakurai et Dave 96) made strenuous efforts to compose the cells into maximal volumes. The composition step starts from a cell and keeps adding cells adjacent to each other, as shown in Figure 1-6-(d). Each maximal volume is then classified into a machining feature through graph pattern matching. Sakurai and Dave were successful to some extent in avoiding awkward machining feature models. However, the resulting maximal volumes may often be unnecessarily complex and awkward in shape, as pointed out by Sakurai et al. [45](Sakurai et Dave 96).

Woo and Sakurai [23] improved delta volume decomposition approach in decreasing the number of cells with so call recursive maximal volume decomposition, thus eased the computational complexity of composing a maximal volume from cells. In their approach, a delta volume was first recursively divided into smaller volumes until each of them having a boundary consisting less than 16 faces. A plan playing the role of bisector was either directly selected from the set of boundary faces of the portion of delta volume being in the process of separation, or created passing the middle of the volume portion.



Maximal volume

Figure 1-6 Cell decomposition of the delta volume [11]

These methods have some natural drawbacks: they cannot directly be used to generate machining features. The generated volumetric forms have to be converted into machining features. These methods are computationally expensive and cannot always guarantee the generation of the correct set of machining features. Last but not least important, all methods are applicable to prismatic geometry volume only.

1.1.3 Efforts in feature classification and standardization

For the purpose to exchange feature data between systems, many efforts have been devoted. Wang and Chang [49] defined features on the base of the shape of the cutting tool and the cutting trajectory. Similarly, Vandenbrande and Requicha [14] classified volumetric machining features in terms of tool swept volumes.

For the same purpose, International Standard Organization (ISO) has lunched ISO 10303, known as STEP (Standard for the Exchange of Product model data). STEP provides a standardized means for the representation of product data to be exchanged between different CAD systems or sharing by different product life-cycle application programs.

The definition of geometric features is found in STEP-AP224 for the process planning in manufacturing of mechanical parts. The features are classified into a fixed number of classes using standard terminology (feature and parameter names). The parameterization is hard-coded. Global coordinates are used to represent the position of feature. In AP224, features are represented in an implicit way or an explicit way. Shape characteristics such as profiles, slot end conditions, pocket bottom conditions etc. are adopted for implicit feature representation. The explicit representation describes the face set of a feature, no explicit topological, geometric or parametric relationships were contained in the set. AP224 translators are not commonly available in commercial systems because it provides no means for modeling general user-defined or special purpose features [11]. Some important factors, such as constraints, association between parameters and explicit geometry, and feature relations or hierarchy, are not considered in the feature definition.

1.1.4 Limitations

Feature recognition (including volume decomposition) algorithms reported in literatures mentioned above have one or more limitations listed below:

1) Geometric information (volume and surface) is the only factor considered for feature recognition, issues related to machining are not mentioned;
2) Features are extracted separately, spatial relationship between them is not considered. This drawback makes it even impossible for a system to sequence features in a list properly.

3) Without taking stock dimensions into account, all features are constrained within the space bounded by the convex hull of the part.

Feature based approaches recognize only 2.5D features, they failed to recognize machining area bounded with curved faces adjacent one with each other.

1.2 CAPP

Process planning is an activity, which determines appropriate procedures to transform a raw material into final product. In manufacturing industry, the task of process planning mainly consists of determining the usage of available resources, such as machine tools, work holding devices, cutting tools, generating operation sequence, determining machining parameters (i.e., cutting speed, feed rate, depth of cut) and auxiliary functions.

CAPP systems were developed to bridge design and manufacturing on the base of computer software, filling the existing gap between computer aided design (CAD) and computer aided manufacturing (CAM), and also to provide data usable to material requirement planning (MRP) for determining standard and optimized production routes [43].

It is noticed that there exist two outstanding issues in creating a process planning model. One is automatic generation of the operation oriented part model from a CAD system, and the other is modeling machining process sequences. The later one is more difficult because it is not only related with the declarative process information, such as part geometry, tooling, machines, and technological requirements, but also has time-dependency associated with the order of the operations [50].

The process planning problem has only been partially analyzed in many research studies, among which the following stand out as pointed out by Ciruana [51]: (a) joining process planning with the part design [50], (b) improving the choice of machining process parameters for cylindrical parts [52] or for prismatic parts [53, 54], or finally (c) optimizing the sequence of operations [55]. Researches that focus on operation sequencing and tool selection, which are two issues discussed in this dissertation, are reviewed in the following sub sections.

1.2.1 Operation Sequencing

Early process planning systems sequence machining operations on the feature base. Typically, four steps were taken to generate machining operation sequences from the CAD model of a part, Step 1, converting the geometry of a part to a set of machining features, which still are merely geometric representations; Step 2, selecting a machining operation for each feature; Step 3, determining setups based on selected machine and fixtures; Step 4, generating detail machining operation sequences. However, sequence for removing features on a global level is not handled systematically in early researches [50].

Sometime later, results on solving the global operation sequencing problem efficiently were reported. Petri-nets were introduced in non-linear process planning to represent the precedence constraints of operations [56]. The reported systems have their advantages in easing integration with production scheduling functions and the generation of alternative operation sequences. In 1993, Gu and Zhang [57] developed an approach that simplified the dependency of constraints by separating the constraints in three groups, each associated with operations, setups, and machining conditions respectively. The main drawbacks of this kind approaches is that the search space expands rapidly as the number of features increase, there is an exponential relationship between them. In the following years, Gu et al. [50] developed another method based on the concept of feature prioritization for solving operation sequencing problem more efficiently. This approach focus efforts on generating operation sequence for important features, and it proved to be very effective and resulting in improved sequencing efficiency. The approaches proposed by Deja et al [58] is limited by the feature precedence matrix which has to be built manually for each part. Similarly, Nallakumarasamy’s approach needs making a operations procedence graph manually [59].

Seeking for optimal machining operation sequence has been an attractive topic in CAPP research. Mathematic tools employed including genetic algorithm (GA) [60], semantic net [61], and simulated annealing (SA) [62]. Various objective functions were developed for different minimization purposes, including mainly the less machining cost, the less time of production, the less number of setups, etc.. However, due to not taking into account the management variables, such as status of raw material, availability of machines, workers etc., the so called optimal machining process plans provided by existing CAPP systems may not be an optimal one while investigated from the production perspective, as they could make some machines overloaded, or under used, and cause subsequent bottle-necking. Thus many people accept that at this moment it is better to let the optimal task be done by production rescheduling. So it is recommended to generate more than one machining operations if the CAPP system does not take into account the production management restrictions. [63].

All these operation sequencing approaches have at least difficulty in relocating the tool access point as well as tool parameters for each feature while the order of operations are changed to get alternative sequences.

1.2.2 Tool and Machine Selection

Tool selection is of particular importance in process planning, since tooling related issues affect most aspects of process planning [64]. Feature geometry is considered as the most important constraint to determining tool geometry and parameters. As machining features form containers for manufacturing and technological information [65], relating tool selection with machining features is required for the vertical integration of tooling methods with design and process planning.

Maropoulos and Baker [66] pointed out that in the aggregate stage of process planning, although features are defined in the early phase of design and thus possess minimal product information, they can still contain adequate manufacturing data to enable comprehensive tool selection. His approach is dedicated to presenting a framework for early tool selection with minimal data, and providing the background necessary to machining time estimation. This purpose was achieved in two steps. The first step is to generate the feature model of a part, as other feature based approaches did. In their approach, a feature model is determined through three actions: creating, validating, and extracting alternative features. The second step is taken to identify the operations necessary to producing the part, and the obtained information is then used in selecting cutting tools. The disadvantages of their approach are 1) limited capacity in feature representation, parameter descriptions can only express simple regular features. 2) Access directions are limited, the positive and negative directions of coordinate system axes. Nevertheless, their ideas concerning operation selection, tool selection and common parameters for operation description are useful to the research of this dissertation.

Sormaz and Khosknevis [67] developed a tool and machine selection approach based on an object-oriented knowledge-based system. Tool types are first selected according to general guideline rules. Then a few of the tool types were eliminated as constraints imposed by the machine and the operation were further considered. This system is an essential part of the 3I-PP system (Intelligent Integrated Incremental Process Planning System). The authors addressed the need for “ideal tool selection”, but failed to explain clearly how that is done.

Usher and Pernandes [68] presented an approach to generating and sequence alternative tool sets for a process plan. It is reported that their approach has been used to develop the OATS system and has been tested as a module of the PARIS dynamic planning system [69]. Both geometrical and technical requirements were taken into account in the approach. Manufacturing meaningful information, such as batch size, material, feature dimension, surface finish, and tolerance etc. were considered for tool selection, but one of the most important aspects, tool approach direction, was not mentioned.

It is noticed that all these researches are based on features and they do not take into account of the need for multi-axis machine to produce the part. Some of them considered the transportation of a part between different machines, or the requirement of multi setups, but failed to consider the case that, for some complex part, even in one setup several machining operations have to be performed at least with a 4-, or even 5-axis position machine. Shabaka et al [70] developed an approach to select machine type and configuration as required by the features of a part. In their model each feature is machinable from one direction.

1.3 CAD/CAM Integration

In the early years of CAD/CAM research, many approaches were developed to solve either feature recognition or process planning as if they were completely separate activities [4]. Later, during the 1990s and more recently, many researchers began to acknowledge the tight relationship between feature recognition and process planning. Systems incorporating feature extraction and process planning [25, 71-73], or integrating design-by-features and process planning were reported. Some of these approaches take the view that a feature extractor should not be required to interpret models; this is the job of a manufacturing planner. Some researchers believe that a feature extractor should generate all valid interpretations to get a more close-optimal process plan. It is a logical consequence that the problem of finding the best interpretation is the problem of finding the optimal plan. Gupta and Nau [74] realized that generating all alternative interpretations is a time-ordering constraint problem. Some approaches were developed to find high quality solutions, if not really optimal operation plans or feature representations [25].

Gupta and Nau [74] introduced the concept of delta volume, and defined that feature-based manufacturing (FBM) is an interpretation of the delta volume as a set of machining features. Some limitations exist in their approach. First, parametrical definition is not applicable to non-linear swept features; and second, there are a lot of exceptions that break the accessibility constraint; For example, in their approach the machining parameters of a pocket feature *p* was defined as $param(p) = \{p_d, v, e, l\}$, where p_d is the starting point, *v* is the unit orientation vector, *e* is the edge loop of the pocket, and *l* is the depth of the pocket. This definition is adequate for a straight swept pocket. Gupta's parameter representation cannot represent a draft pocket, where the edge loops for each cutting layer are approximated but not equivalent. In fact, there exist enormous varieties of irregular pockets that cannot be represented by means of parameters; no matter how complex a parameter structure could be constructed, exceptions can be easily found. A better way is to use the geometry information about the volumetric status of an irregular pocket. In this research we prefer using the Brep solid model for representing an irregular pocket. With the solid model, we can not only determine the start point and end point (the depth), but also edge loops, even nonequivalent, for all cutting layers, furthermore, it will become much more easy to change cutter orientation to remove the portion which is inaccessible from the direction perpendicular to the top or bottom face of the pocket. Examples show the failure of their accessibility constraint are: a hole on a slant face should be drilled before the slant face was created [75]; two axis coincidental separated holes may be required to be drilled in the same operation before the volume separating them is removed.

1.4 Synthesis and objectives

Complete integration of CAD and CAM systems depends heavily on the innovation in CAPP. Machining process planning is a procedure for determining the operations necessary to transform raw material or partially finished workpiece from one state to another. The consequence of process planning is an instruction of machining operation sequence. The kernel of the plan is a set of procedures (steps) to be carried out for achieving the objectives of the plan. There always exist alternative operation sequences that result in the same part.

A machining process plan consists of the specifications for a set of machining operations. The specification for an individual cutting operation includes at least two types of data [76, 77]:

1) Description of the operation, including the type of operation, the specific type and parameters of a tool to be used, machining parameters such as feed rate, spindle speed, radial cutting depth and axial cutting depth, etc.

2) Specifications of the geometry of the material to be removed, including shape, location, orientation, and special relationship with other related removal volumes. All geometric specifications are represented in the local reference system which is attached to the stock or workpiece.

CAPP accomplishes the complex process planning task by considering the total operation as an incorporated system, so that the individual steps involved in machining each part are coordinated and executed efficiently and reliably [78]. The key role that a CAPP system plays in the scenario of computer integrated manufacturing (CIM) is an interpreter of CAD description of a part to CAM understandable representation.

1.4.1 Limitations of existing systems and research

Although sophisticated CAD/CAM systems have been developed, many manual interactions are still needed for defining machining operation processes, and thus a bottle-neck phenomenon occurs while the product data flows from product engineering to manufacturing engineering. Concurrent engineering and web based product development environments need to overcome this bottle-neck by bringing CAD-CAM integration further forward, increasing the productivity of process planners and CAM programmers, and facilitating communication between design and manufacturing departments.

During more than two decades, enormous efforts put on CAD, CAM, and CAPP have brought about evident improvement in description translation by mean of feature recognition, feature based design, and machining feature library building. With the survey of some of the previous research work mentioned above, it is noticed that there are one or more of the following limitations in each of these literatures.

1) Feature is a commonly accepted concept, although the definition of feature is general for any shape, only swept features were reported in the literatures. Features found in

the previous researches, either defined by parameter or recognized with one of the approaches mentioned above, are of the shape that can be generated by sweeping a 2D profile along a trajectory, under the condition that at every point on the path the profile has to be perpendicular to the tangent vector of the trajectory. As being mentioned above, the most important advantage of using swept features is that they can be defined parametrically with less difficulty. However swept features fail to represent complex volumetric shapes of the removable material that exist universally in aerospace workpieces. To be distinguished from swept feature, in this dissertation “sub delta volume” is adopted as the term for describing the volumes of removable material. The concept of sub delta volume is explained in detail in section 2.3.2.

2) Features extracted from the CAD model of a part are treated isolatedly. While discussing intersection, only geometric crossing was investigated in detail. The situation in which cutting one feature may lead to removing a portion of another feature, known as machining intersection, was nearly untouched. Even when the accessibility of features was discussed by some researchers in the alternative process plan issue, relationships between features were not well analyzed. As a result, the possibility of intersectional machining was not mentioned in previously published researches. In practice the following situation may usually happen. One process planner specifies an operation sequence, follow which two features are supposed to be machined by two completely separated operations, even though the two features are adjacent. While in an alternative plan made by another process planner, operations for removing the same features will be arranged in an inversed order as referred to the prior plan. In the later plan, a portion of one of the features, or the whole feature may be removed away by the precedent operation. In this case, we need not only take into account of the change of the feature shape in the sequence, but, much more important, the change of tool and machining parameters, as well as the access position of the tool.

3) Constrains on accessibility of two vertically adjacent features prohibit the existing algorithms from generating alternative process plan that may usually be made by a process planner according to his experiences.

4) Constraints on TAD led the feature method to be inapplicable to some kinds of volume that require to be accessed from a direction neither perpendicular nor parallel to the top/bottom face. This kind of volume exists commonly in aerospace structural parts.

5) Lacks of volumetric consideration, models presented in the literatures do not distinguish tool parameters more sufficient for roughing from those for face machining.

6) The existing approaches give job to programmers to define the TAD and access point for each feature. Divers results can get and any change in process sequence requires the whole job to be redone completely.

7) To construct alternative machining process plans, all features have to be regenerated again, even when the majority of the features remain unchanged for various plans. This will seriously increase the computational burden to search for optimal operation sequence under certain criteria. The computational pressure can be multiplied if plural optimization objectives are being sought.

1.4.2 Dissertation Scope and Problem Definition

In order to improve existing CAD/CAM system with more powerful functions in transforming the design model of complex part into the machining model, and generating alternative machining operation sequences, so that to facilitate the job and increase the productivity of process planner and CAM programmer, a new systematic approaches is going to be developed. These approaches should be capable of :

1) Developing new approaches capable of generating manufacturing meaningful geometric representations of complex parts directly from the CAD model combined with a given 3D stock model;

2) Generating volumes of removable material with information necessary for automatic operation sequencing, and ensure the flexibility of changing the order of some of the machining operations as a process planner may do according to his experience;

3) Determining basic tool parameters and access positions in accordance with specific machining operation sequence;

Aerospace structural parts are good examples of complex design model, which require all the improvement listed above. Our research is based on analyzing the characteristics of aerospace structural part. Usually an airframe structural part is machined out from a block of aluminum alloy by a number of mechanical cutting processes with a CNC milling machine. Two setups are normally needed to complete the structural part, one for removing most of the delta volume and create most of part faces, the other for machining the rest sections of the part and cut down the part from the residue of the stock. The complete machining process is divided into two or three stages: roughing, semi-finishing, and finish, some part may not require semi-finishing. Most roughing operations remove a certain volume of material of a prismatic geometry, and each of these operations is actually performed in 2.5D way, in which the volume of material is removed layer by layer of the thickness as determined by vertical cutting depth. In the machining process, a workpiece is usually fixed in one of two ways; held by a vise or pressured firmly against a supporting face by clamps.

The problem is how to integrate all the functions listed above. Given an airframe structural part, together with the rectangular stock from which the part will be produced, how to find out all milling operations for roughing processes, determine tool parameters for each operation, generate acceptable operation sequences, determine the access position of a tool for each operation in the sequences, determine required number of axes of a machine tool, as well as required angular and linear move capacities of a machine?

In this dissertation, we limit our research within the following scope: considering only 2.5D milling operations for roughing an aerospace structural part from a rectangular prismatic stock. The 2.5D condition restricts only the way in which tool path will be created. As several operations are accomplished in one setup, a machine tool capable of performing multiple axis machining may be required to perform consecutive surfacing operations. This dissertation view the problem from the perspective of geometry, thus technology parameters such as feed rate, cutting depth, spindle rate, tool material are not taken into account. It is supposed that there is enough margin between the stock boundary and the part boundary thus warrants sufficient space for passing the tool without collection with the fixtures.

1.4.3 Research Objectives

This dissertation aims to develop systematic approaches for transforming the representation of complex part from its CAD model to a machining meaningful geometric model under the circumstance of a given stock, generating alternative machining operation sequences with the flexibility that the order of operations can be changed by a process planner based on his/her expert knowledge, determining tool type, basic tool parameters and tool access position for each operation, determining the number of axes required for a machine tool to be able to complete as many as possible operations in one setup. To make the developed approaches applicable to aerospace structural part, the following functions are essential:

1) Determining tool parameters according to both surface and volumetric information;
2) Determining the machinability of a part by means of milling operations;
3) Determining the machine type for performing as many as possible operations in one setup, to take the advantage of multi axis machining capacity of a machine tool.

The generated geometry CAM models should contain all geometric information necessary for building finishing models of a part.

CHAPTER 2 CONCEPTION AND METHODOLOGY

A CAD system is used to define the nominal geometry and tolerance, material and other attributes of a part. A CAD model is the digital expression of a part, including all geometric and technical information about a part. To produce a part, a manufacturing engineer must take all specifications of the part into account. A CAM system is developed mainly for the task of generating tool path for a specific machining operation. Of course, machining parameters are also defined with a CAM system. The input data of a CAM system is not the specification of a part, but the descriptions of series operations. The full description of a cutting operation necessary for generating tool path and machining parameters including at least four categories of information: type of operation, geometric definition of the area or surface to be cut, type and geometric data of the cutting tool to be used, and the cutting parameters. This dissertation does not discuss the cutting parameter determination and optimization. A lot of researches on this topic can be found in the published literatures [51, 79-85]. In a route sheet (manufacturing operation sequence), all the machining parameters such as the cutting speed, the cutting depth, the spindle rate, etc. are specified. Although it is one of the most complex tasks to optimize machining parameters [86], in practice, the parameter determination process is not a time consuming job. Today, in most workshops, the cutting parameters applied on a machine tool in cutting processes are not the same as those specified in the route sheet. They are very often changed by a machine tool operator according to his experience.

This dissertation is concentrated mainly on geometric data transformation because it is considered the least powerful link in the chain of CAD-CAM integration. A great gain could be obtained if all processes concerning geometric conversion were completed automatically. Aiming at this achievement, the research of this dissertation is developed pursuing the following methodology.

2.1 Assumptions

Our approaches are based on the following assumptions.

Assumption 1: A part is machined from a rectangular prismatic stock, and the digital representations of them are 3-manifold with boundary (a compact entity dividing the Euclidian space into two parts: its internal and external spaces). The total volume of material to be removed from the stock to produce the part is represented digitally as the result of regularized Boolean subtraction of the part from the stock. The difference of the subtraction is also a 3-manifold with boundary. Thus further set operations can be applied on the difference to separate it into smaller 3-manifolds with boundaries.

Assumption 2: The part has at least one planar face that is large enough to be used as the support face, and one of the stock faces has been machined in accordance with the finish requirement of the supporting planar face. During machining the support face is parallel to the top face of the machine tool. In other words, the supporting face is coincident with one of the reference faces, usually the XY plane.

Assumption 3: In roughing operation the deformation of the workpiece is ignorable. For cutting efficiency, a cutter of the diameter as big as possible can be chosen for a specific roughing operation. Each individual operation can be performed with tools of different types and dimensions. The bigger the diameter of a tool is, the more efficient the operation can be performed.

Assumption 4: The digital model of a part and stock, as well as derivative smaller volumes are advanced Brep model, so that the functions representing the surfaces and curves corresponding to faces and edges of these entities can be called and operated to get mathematical results corresponding to specific questions.

2.2 Methodology

As mentioned above, full description of machining operation is the input of CAM system. Among the three kinds of geometric information, the definition of machining area is the primary, the other two sorts of data: operation type and cutter characteristics are determine on the base of machining area geometry. Furthermore, the special geometric relationship between machining areas can be used as constraints to sequencing operations. For this reason, our research is developed pursuing the strategy of three main steps: geometry decomposition, mapping geometries with machining operations, generating alternative operation sequences.

2.2.1 Geometry decomposition

The first step is to determine the geometries of all individual machining areas. Since the variety and irregularity of the shape of machining area of aerospace structural part, the existing approaches have difficulties in generating machining areas, furthermore, the existing approaches do not associate tool access position and tool parameters with features, they are lack of the function to determine the relationship between machining areas, etc. This research needs to begin with developing a new approach for machining area determination applicable to complex aerospace structural part.

Delta volume[87], a concept well accepted by researchers in this field, is adopted in this dissertation. Delta volume is the total volume of material has to be removed from a stock to produce a part. Another well-known concept “feature”, representing a partial of delta volume which can be generated by sweeping a planar section along a guide curve, is not utilized in our work, because, in producing an aerospace structure part, 1) very often some areas of removable volume of material are not swept volumes, thus they cannot be taken as features; 2) the criteria of removal using one cutting operation is not applicable to certain removable volumes of material by milling. Very often, certain volumes recognized as one individual machinable area may actually not be removable in one operation. To avoid being confused with the conventional feature concepts, in this research, sub delta volumes (SDV), which are portions of delta volume obtained by decomposition processes, are used to represent a single volumetric area of removable material.

Based on the characteristics of aerospace structural part, and knowledge of basic machining procedures of milling operation, the proposed approach is developed to obtain individual machining areas in a top-down sequence. Although in practice a part may be machined from interior towards exterior, to facilitate the decomposition algorithm, a delta volume is decomposed from surrounding material towards the center.

Different from other volumetric approaches presented in the references [23, 25, 37, 45, 88-91], which deal with the swept features or the prismatic features, our approaches are dedicated to complex part with areas that cannot be described as features of conventional meaning. The important aspects of the developed approaches are summarized as follows:

1) Original delta volume: Our approaches are based on the original delta volume which is a 3-manufold with boundary. The original delta volume is obtained by regular subtraction of the part from the specified stock. Both of the stock and the original delta volume are 3-manifolds with boundaries. As the part is machined from the stock, so that the stock is a proper superset of the part.

2) Splitting face creation: To our knowledge this is the first time that a splitting face is introduced for volume decomposition. A splitting face is the boundary of partial half space. It separates a specific volume from the remaining portion of the delta volume so that a sub delta volume is created. A splitting face can be of any shape, including a union of faces. The sub delta volume is the volume on its positive side. For details, please see sections 3.3.5 through 3.3.9.

3) Face classification: The classification is made from the point of view of machining. Boundary faces of a sub delta volume are classified into two groups: open faces and close faces. Open faces are removable, and close faces have to be created by surfacing. Based on the classification access faces can be selected from open faces, an access point can be associated with each access face, so that to know, on the boundary of a sub delta volume, where a tool can start cutting the material.

4) Recursive subtraction: The decomposition is carried out in a recursive subtraction process. In each step a new sub delta volume is separated from current delta volume which is the portion of original delta volume remained after the preceding separations. Then the delta volume is updated by subtracting the new sub delta volume from the current delta volume. As the result of the subtraction a new remained delta volume is obtained, which is the object to be decomposed in the next step. The decomposition process terminates when the difference of subtraction equals to zero.

After decomposition machining attributes such as access face, required tool parameters etc. are assigned to sub delta volumes and their related faces.

2.2.2 Manufacturing attribute determination

Manufacturing attributes of a removable volume of material (named sub delta volume in this dissertation) considered in this research include access face, access point, access direction, and required tool type and parameters. The proposed approaches are developed to identify all possible access faces of a sub delta volume, determine an access point and an access direction for each access face correspondingly, and specify required tool type and parameters. Meanwhile functions to analyze machinability of a sub delta volume are also added to these approaches.

2.2.3 Machining operation selection

Instead of creating operation patterns and then select an operation by mapping the sub delta volume with the patterns, in the proposed approaches each sub delta volume is assigned an operation according to the configuration of its boundary faces. Theorems are developed to verify face configurations of sub delta volumes.

2.2.4 Operation sequencing

One of the main purposes of operation sequencing is to give a list of all operations fulfilling the production of a part. In this research we consider only roughing operations. Most often in practice alternative sequences get the same results. Operators often change the operation sequence according to their experiences. Rules are proposed in this research with the consideration of spatial adjacency and precedence of access from given directions.

Although it is proposed to sequence operations in such a way that, in a given direction, the “higher” special position of a sub delta volume, the higher position of the corresponding operation in the list, for the flexibility of changing operation sequence, the algorithm is developed to recalculate tool parameters according to a given operation sequence. After re-sequencing operations, the shape of a SDV may need to be changed if it is situated specially on the “top” of a SDV to be machined before. To avoid machining the same volume twice, the shape of the SDV on the “top’’ has to be redefined by subtracting the intersection of its volume and the volume swept by the tool.

The proposed approaches have the advantages of tool re-positioning and tool parameters re-defining according to changes made to an operation sequence.

2.3 Terminology

For the convenience of description in the following sections, some of the basic terminologies introduced or adopted in this dissertation are defined. In our definitions new attributes are assigned to some commonly known terms, such as delta volume, accessibility. Completely new concepts such as sub delta volume, splitting face etc. are defined as well.

2.3.1 Original delta volume and current delta volume

Delta volume is a term well accepted in CAD/CAM field. It is defined as the total volume of material that has to be removed from a piece of row material to produce a part. Geometrically delta volume equals to the volumetric difference between a piece of row material and a part.

In this research, approaches were developed to decompose delta volume into smaller portions, each of which is separated from the main volume by one decomposition operation separates. The residual volume subjected to consecutive decompositions represents the remaining volume of material to be removed. For express convenience, we define the residual volume as “current delta volume” (current DV) with respect to the next decomposition, and the delta volume at its initial state is named “original delta volume” (original DV). In other words, each decomposition operation has a specific current delta volume as the decomposition object. The shape and volume of current delta volumes are changing as the decomposition processes are applied to the volume of removable material one after another.

Delta volume is a 3-manifold with boundary, see Figure 2-1. Most of the interior of DV is removed by roughing machining, the rest of the interior and the boundary common with the boundary of a part are machined by semi-finishing or finishing. In this research, the shape of delta volume is progressively altered as the machining operations are accomplished one after another. The original delta volume denoted DV[0] is defined as the regularized Boolean subtraction of the part from the specified stock. As mentioned in section 2.2.1, the stock must be a proper superset of the part.

More preciously, given the 3D models of a part P and a rectangular stock S, with the criteria that P is a proper subset of S, $P \subset S$, the original delta volume DV[0], representing the total volume of material that have to be removed from S to produce P, is defined as the complement of P in S, that is

$$DV[0]= S-^{*} P \tag{2.1}$$

where $-^{*}$ denotes the regularized Boolean subtraction.

Denote DV[i] the current delta volume before the i-th decomposition operation, in the following sections of this thesis, without causing confusion, "delta volume DV[i]" means the same as "current delta volume DV[i]". The mathematical definition of current delta volume DV[i] will be given in the next subsection.

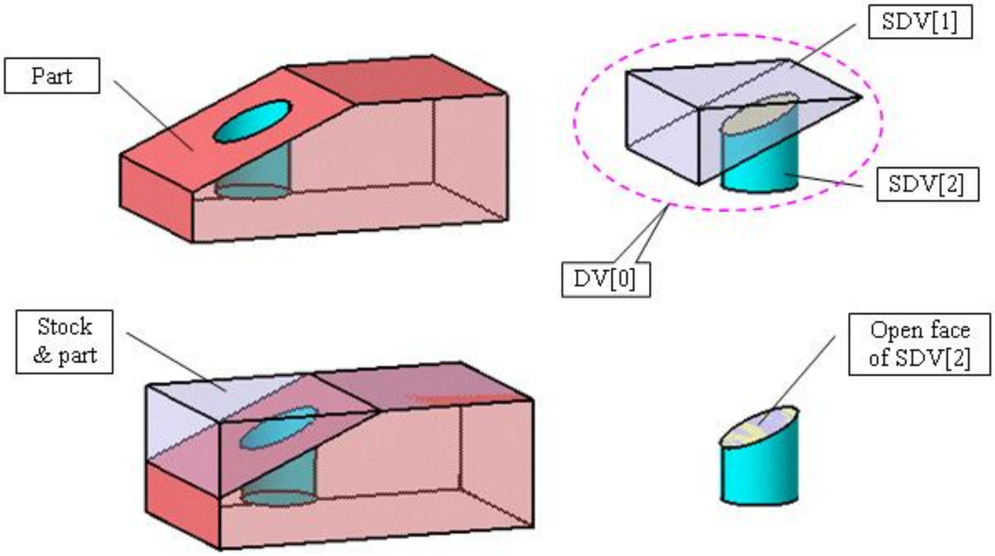


Figure 2-1 DV, SDV and open face

2.3.2 Sub delta volume

Sub delta volume, denoted SDV, is a portion of DV, also a 3-manifold with boundary. In this dissertation, a SDV is obtained by splitting a DV into two portions. The face used to perform the splitting is defined as a splitting face. The portion situating on the positive side of the splitting face is the SDV to be created.

As the machining process is carried out in subsequent operations, following similar progressive processes the original delta volume DV[0] is decomposed into sub delta volumes by removing a single sub delta volume SDV at a time. Set **DV** is a collection of delta volumes $\mathbf{DV} = \{\mathrm{DV[i]}, 0 \le \mathrm{i} \le n\}$, each element DV[i] of set **DV** represents a residue of DV[0] as if a series of SDVs have been removed by machining operations. Set **SDV** is used to represent the collection of sub delta volumes $\mathbf{SDV} = \{\text{SDV[i]}, 0 \le \text{i} \le m\}$, where SDV[i] denotes the volume of material to be removed by one machining operation, at this stage it can be understood that SDV[i] will be removed by the *i*-th operation. Hereafter, if not confusion will be caused, DV and SDV are also used to represent a single delta volume and a single sub delta volume respectively. The delta volume DV[i] is uniquely defined by

$$\text{DV[i]} = \text{DV[0]} -^{*} \left(\bigcup_{j=1}^{\text{i}} \text{SDV}[j] \right), \ \text{i} \ge 1 \tag{2.2}$$

where $-^{*}$ signifies regularized Boolean subtraction.

Equation (2.2) can be written in a recursive style by denoting

$$\text{DV[i} - 1] = \text{DV[0]} -^{*} \left(\bigcup_{j=1}^{\text{i}} \text{SDV[i} - 1] \right), \ \text{i} \ge 1 \tag{2.3}$$

and substituting in the right side of (2.2), we get

$$\text{DV[i]} = \text{DV[i} - 1] -^{*} \text{SDV[i]}, \ \text{i} \ge 1 \tag{2.4}$$

or

$$\text{DV[i-1]} = \text{DV[i]} \bigcup \text{SDV[i]}, \ \text{i} \ge 1 \tag{2.5}$$

Equation (2.4) can be interpreted as DV[i] is the remaining portion of DV[i-1] after SDV[i] is removed, thus DV[i] can be considered as the residue of DV[i-1]. As DV[i] and SDV[i] come from DV[i-1] by splitting DV[i-1] into two divisions, they are considered as the children of DV[i-1]. The decomposition is carried out in a recursive way. In each step, the input is the residue DV[i] of DV[i-1], the output is the lower generation, DV[i+1] and SDV[i+1].

For the purpose of machining operation sequencing, a sub delta volume SDV[i] should have machining attributes: open or closed, and the adjacency to other SDV.

2.3.3 Delta volume decomposition

Delta volume decomposition is the process used to subdivide a delta volume DV into sub delta volumes, denoted SDVs, with a finite number of Boolean operations, under specific strategies that guide and constrain decomposition procedures. In this dissertation, the following rules are applied to guide the decomposition:

1) Setup dependent: The proposed procedure of volume decomposition is similar to a machining operation sequence. It is proposed to select the largest planar face of a part as the supporting face. Changes to the posture of the part and/or the stock may lead to different results of decomposition. It is suggested to position the part and stock in the position that they will most possibly be posted in practice.

2) Facing first: As facing is usually the first machining operation performed in milling roughing, the SDV for facing should naturally be determined first in the decomposition.

3) Convex volume: Generally it is desired to decompose delta volume into convex SDVs so that to facilitate tool path, to avoid limiting the tool diameter by evidently narrowed areas, thus to enable selecting bigger tool to have higher roughing efficiency. Approaches for generating a concave SDV is also developed in this dissertation.

4) Top down: The decomposition procedures are carried out from the top to the bottom as if the part is fixed on the table of a vertical milling machine, so that the SDV in a higher level is determined before those in lower levels.

5) Continuous surrounding: The material continuously surrounding the part from outside should be determined as a single SDV either by splitting or by combining the exterior SDV together. So that the contour of the part can be machined in one operation. In contouring the tool does not need to sweep the whole volume of the SDV, thus to save machining cost.

6) Generated face: In our approach to perform splitting a delta volume it is usually required to create new faces. These faces are call generated faces because they do not belong to the boundary of the original delta volume DV[0]. Generated faces are used as splitting faces to cut off SDVs from corresponding delta volumes. In the boundary of a SDV every face overlapping with a generated face does not belong to any part face.

2.3.4 Open face and closed face

A boundary face of a SDV is open if a tool can pass through the interior of the face without damaging the part. In other words, an open face will be removed during the machining operation. On the other hand, a closed face of a SDV is a face that is completely or partially shared by the part with the SDV. Theoretically a closed cannot be removed by any operation. It will be in fact created by a specific operation. In this dissertation it is not necessary to require an open face be always exposed to the air because another SDV sharing the face and situating "above" it may still exist. For example, an inclined hole in a planar face is usually machined before the face being created, see Figure 2-1.

2.3.5 Adjacency with other SDV

Two sub delta volumes SDV[i] and SDV[j] are adjacent if one boundary face f of SDV[i] is a portion of the boundary face g of SDV[j]. More preciously, given two sub delta volumes SDV[i] and SDV[j], denote ∂(SDV[i]) and ∂(SDV[j]) the set of their boundary faces. If $\exists f \in \partial(\text{SDV[i]})$ and $g \in \partial(\text{SDV[j]})$, such that $f \subseteq g$, then SDV[i] and SDV[j] are adjacent. As we know from the concept of open face and closed face, it is intuitive that two sub delta volumes can be adjacent only at an open face of the SDVs.

2.3.6 Top/bottom and wall face

The top face of a SDV is the first face touched by a tool in the roughing operation applied to the SDV. A top face is an open face which can be penetrated by a tool without damaging the part. Any open face of a SDV can be selected as the top face.

The bottom face of a SDV is the last face touched by the tool in an operation. A bottom face is determined corresponds to a top face. A bottom face can be either an open face or a closed face.

A SDV must have at least an open face and a bottom face as its boundary faces. Other faces between top face and bottom face are defined as wall faces. A wall face can be open or close.

Faces tangent to a bottom face but do not share any edge with corresponding top face, an edge fillet face for example, can be jointed with the bottom face, or treated separately as other faces.

Both top face and bottom face depend on setup position and access direction, as well as wall faces. Their roles can be changed when the same SDV is viewed from different access directions. These terms are employed in the process of delta volume decomposition and operation sequencing.

2.3.7 Milling operation attributes

In this dissertation we define milling operation attributes of an SDV as the information about the position of a tool at the beginning of the operation to remove the SDV, and the requirements on the tool to perform the operation. Required machining parameters should also be included in the attributes, but they will not be discussed in this work. Access face, access point, access direction, tool type and parameters are considered here as attributes necessary to machine a SDV.

2.3.8 Access face

An access face is one of the open faces of a SDV selected as the interface defining geometrically the beginning of an operation. An operation starts at the moment when a tool breaks the access face and gets into the volume enclosed by the face. Access faces are selected from the open faces of a SDV. Denote SDV.f.open and SDV.f.acc the set of open faces and access faces of SDV, we have

$$\mathrm{SDV.f.acc} \subseteq \mathrm{SDV.f.open} \tag{2.6}$$

Criterion (2.6) implies that not all open faces can be access faces. Two of the cases that an open face cannot be an access face are noticed:

a) The face is too close to a part face, measured in its normal vector direction, to allow a tool of reasonable small diameter to pass.
b) In the boundary of a face no edge is long enough to allow a tool of reasonable small diameter to pass through.

2.3.9 Access point

An access point is the point at which the tip of the tool has to be located on the moment to start the machining operation defined to remove the SDV. An access point is determined correspondingly to an access face, it can be a point in the interior or on the boundary of an access face.

2.3.10 Access direction /Tool Approach Direction

Although introduced or employed in several literatures [92] [93] [94] [95], the concept of tool approach direction (TAD), or access direction, machining approach direction as named by some researches, is not clearly defined, and TADs are considered of process sequence sensitivity [93]. Usually a TAD means one of the six directions: $\pm x, \pm y$ and $\pm z$ [96]. It is very often in producing an aeronautical structural part, a tool has to be oriented in the directions differ from these six directions. And it is more convenient to define "access direction" a term that is independent of process sequences.

In this dissertation "tool axis vector" or "access direction vector" is defined to represent the access direction, and hereafter when TAD is used, it means the specific tool axis vector. An access direction vector is a vector having its end point at an access point. It indicates that to start a machining operation the axis of a tool and the center point of tool tip must be coincident with the vector and the end point of the vector respectively.

Intuitively, there are infinitive directions in a half space along which a tool axis can be aligned to roughing a volume of material, excluding removal of some special shape of volumes with dedicated cutters, such as to drill a hole and to make a slot. Usually different TADs are required to remove a volume of material completely. Figure 2-2 illustrates a case in which multi TADs are needed to achieve a better roughing result, (a) the part, (b) the area removable by TAD 1, (c) the area accessible to TAD 2.

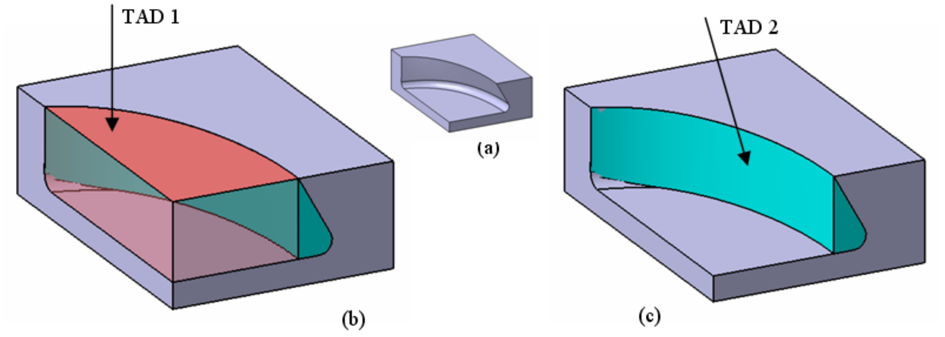


Figure 2-2 Multi TADs required removable volume

In practice, it is also important to discriminate between axial accessibility and side accessibility of an open face. Figure 2-3 shows an example, face 1, 2 and 3 are open face of the removable volume. Among them only face 1 is axial accessible by a tool, face 2 and 3 are side accessible.

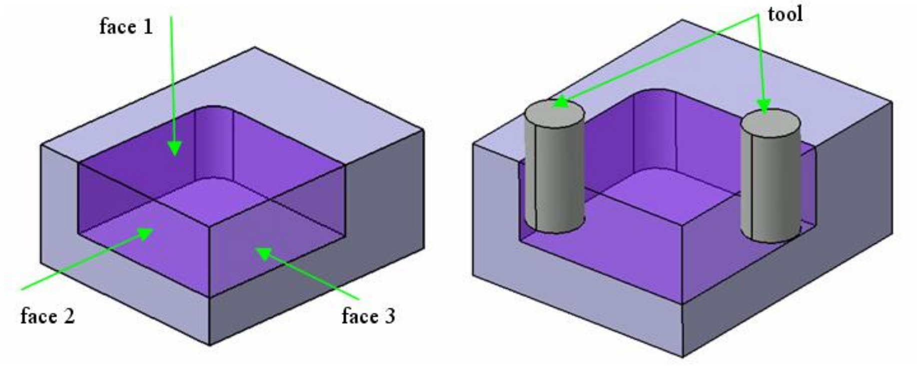


Figure 2-3 Axial accessibility and side accessibility

2.3.11 Operation Sequencing

Operation sequencing is the procedure to arrange all the machining operations for producing a part in an ordered list, so that they are performed one after another. The same set of operations can be sequenced in different ways so that to get different operation lists (alternative operation sequences). Each of the lists represents a specific order of operations that can be executed for producing a part. When machined by milling, the material of each sub delta volume is removed layer by layer, and getting deeper and deeper in the direction that may be parallel, perpendicular or even tilt to the tool axis. So the TAD is an important factor for sequencing operations.

In this work only geometric constraints were taken into account for sequencing machining processes. In addition to the constraints listed in [97], several new criteria were added for order priorities, such as z coordinate priority, axis coincident priority, inner priority, TAD priority, volume priority, etc.

Each ordered list of operations is for operations that can be performed in one setup. For multi-axis milling machining, the capacity of maximum angular and linear displacement of a machine tool is one of the important factors that determine the possibility of one setup machining. For example, if a machine table has a rotation domain of (-180, 180) and the machine head can pivot in the range of $(-\alpha,\alpha)$, then the TADs of operations performable in one setup are limited in the half space with a conic boundary, the aperture of the circular cone is 2α .

Other terms will be explained when their first appear.

2.4 The Contributions

This research contributes the following innovations to CAD/CAM integration.

1) Enriched the concepts related with delta volume decomposition. Proposed the concept of sub delta volume to represent the volume of material removable in one machining operation. The most significant difference between these two concepts is that a form feature is a prismatic volume, while a sub delta volume can be any 3D geometry.
2) Systematic approaches for decomposing a delta volume DV into sub delta volumes SDVs, thus to transfer the CAD model of a complex aerospace structural part into a manufacturing model.
3) Approaches for determining the machining attributes of a SDV presenting in the environment in which exist other SDVs of the part. Tool type and parameters are components of the machining attributes of a SDV.
4) Approaches for determining machinability of a part by means of analyzing the machinability of its SDVs.
5) Approaches for assigning operation to each SDV without patterns of operations.
6) Approaches for generating machining operation sequence, with flexibility to allow a process planner to change the order of operations, capable of re-determining machining attributes of SDVs for alternative operation sequences.
7) Approaches for automatically re-defining the dimensions of a stock and update SDVs in case that the size of a selected stock may be not enough in one or more dimensions.

In the following chapters, our approaches are presented in detail.

CHAPTER 3 DELTA VOLUME DECOMPOSITION

Converting the CAD model of a part into machinable spatial areas is an essential step of CAPP. In this research this conversion is achieved by dividing the total volume of removable material into smaller volumes, each can be removed with a tool parameterized for time efficiency and/or shape fitting from a preferred access direction. As defined in chapter 3, in the case of producing a part P from a stock S by means of milling, the total removable volume of material is defined as initial delta volume DV[0], and every portion separated from DV[0] is defined as a sub delta volume, denoted SDV. This chapter explains our Approaches for decomposing delta volume DV[0] into SDVs.

3.1 Determination of DV[0]

Given the solid models of a part P and a stock S, both are 3-manifolds with boundary. Denote fp_0, fb_0 one of the boundary faces of P and S respectively. Suppose S and P are positioned in such a way that P is completely included in S and fp_0 is overlapped completely with fb_0. Or described mathematically, S is a superset of P, that is

$$\mathrm{P} \bigcap \mathrm{S} = \mathrm{P} \ \wedge \ \mathrm{S} \backslash \mathrm{P} \neq \varnothing \tag{3.1}$$

The concept that fp_0 overlaps completely with fb_0 is defined as $fp_0 \subseteq fb_0$. In this case, face fb_0 is taken as one of datum faces for the machining, and the workpiece is fixed on the machine with this face being parallel to the machine table. By positioning P and S in such a way, the total volume of removable material is determined by

$$\mathrm{DV}[0] = \mathrm{S} -^{*} \mathrm{P}\,. \tag{3.2}$$

where $-^{*}$ denotes the regularized Boolean subtraction.

Figure 3-1 shows the initial delta volume DV[0] of a sample aerospace structure part. DV[0] consists of geometric information necessary to determining sub delta volumes SDVs. Each SDV represents volume of a specific spatial area unambiguously defined in a reference system. It can be removed with one machining operation. Our approaches are based on the Boundary Representation (Brep) of a part P, a stock S, and corresponding DV[0] obtained from equation (3.2). The main objectives of volume decomposition can be described as:

1) Determine the geometry of individual SDVs, each of them is removable by a single machining operation if it is machinable;

2) Generate information that is meaningful to machining, such as open face that is accessible to a cutting tool, adjacency relationships between SDVs, etc. These kinds of information are necessary for determining the machining attributes of the SDV, analyzing machinability and generating an operation sequence.

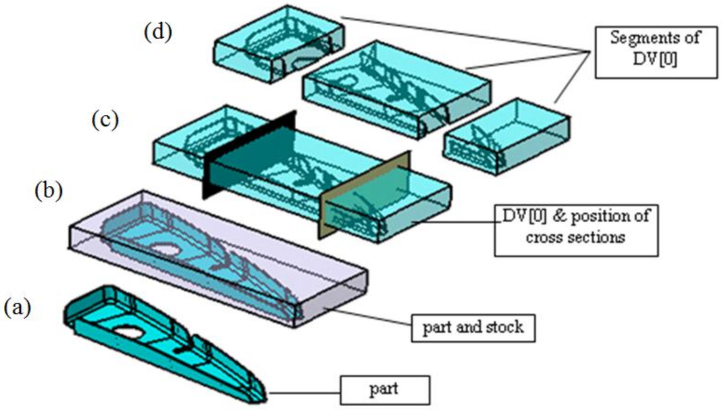


Figure 3-1 Sample part I and delta volume DV[0]

For seeing the inside of DV[0], the delta volume is cut into three segments with two cross sections, as illustrated in Figure 3-1 (c) and (d).

3.2 Decomposition strategies

The decomposition is based on the Brep model of delta volume and it is carried out by recursively separating a portion of volume from current delta volume with a splitting face. The strategies of decomposition can be summarized as follows.

1) Each splitting creates one SDV.

2) A splitting face is not necessarily passing any boundary element of current DV.

3) Decomposition is carried out from the outside towards the inside of DV[0].

4) A splitting face can be a single face or the union of several faces including some curved boundary faces of a DV.

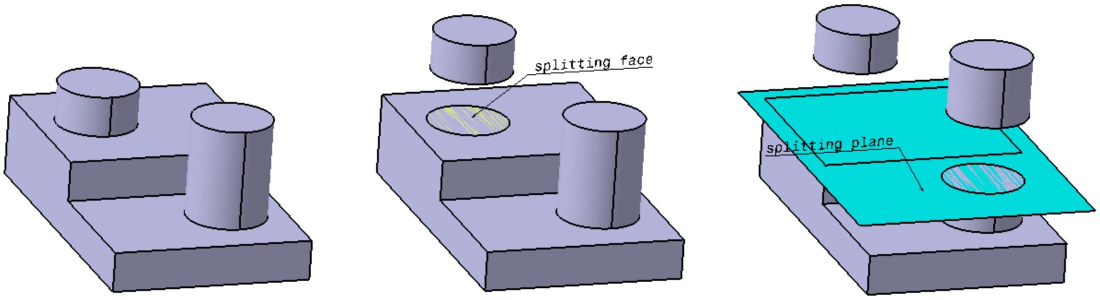
5) Every extruded volume should be separated as a SDV.

6) Decomposition terminates when no volume extruding from any face of current DV can be found.

Further explanations to the above strategies are given in the following sub sections.

3.2.1 One SDV by one splitting

It is very often that cutting a delta volume can result in more than two separated volumes if the splitting face is not properly formed. Decomposing multi separated volumes enhances the complexity in algorithms, thus decreases the computational efficiency of decomposition. Moreover, it may result SDVs of unexpected geometry form. That is why we insist on getting one SDV by one splitting, see Figure 3-2 (b). To obtain one SDV in each partitioning, the slicing operation has to generate a 3-manifold SDV and the other portion of the split volume has to remain a 3-manifold. That is why the area of a splitting face is controlled so that it does not slice two islands at the same time. This is guaranteed by forming a splitting face intersecting only with the boundary of one SDV. Figure 3-2 (c) illustrates an example of two island split by a plane. The splitting results two SDVs which need to be distinguished one from the other. Furthermore, the bottom part of the island in the front needs to be determined as another SDV.



(a) DV

(b) split by a face

(c) split by a plane

Figure 3-2 Difference between splitting by a face and a plane

3.2.2 Location of a splitting face

In most cases a splitting face is generated passing through at least one vertex or one edge of the DV under decomposition. But passing through an existing vertex or edge is not a necessity for a splitting face. For example, in the case that a facing volume has to be determined above an extrusive curved face, as shown in Figure 3-3, the splitting face has to pass the summit of the curved boundary face of DV[0], by any vertex. In Figure 3-3, for better visualization the coincident point on the part face was highlighted instead of marking the summit of the boundary face of DV[0].

3.2.3 From outside towards inside

In our approaches, the decomposition is carried out from the outside towards the inside of the DV[0]. The reason for the outside priority is that at each step the open face of current DV can be used as reference for determining the next splitting face. The SDVs with at least a portion of a stock boundary face have the priority of being created before those that are not enclosed by any stock boundary, as well as the facing SDV is created first. One exception is that some of the SDVs having faces overlapping with the bottom face of the stock may need to be separated later, which obey the top down priority. When the stock boundary is completely removed, the current DV enclosed merely by created faces and faces defining the shape of the part is defined as an interior DV. The volume separated from an interior DV is mentioned as an interior SDV.

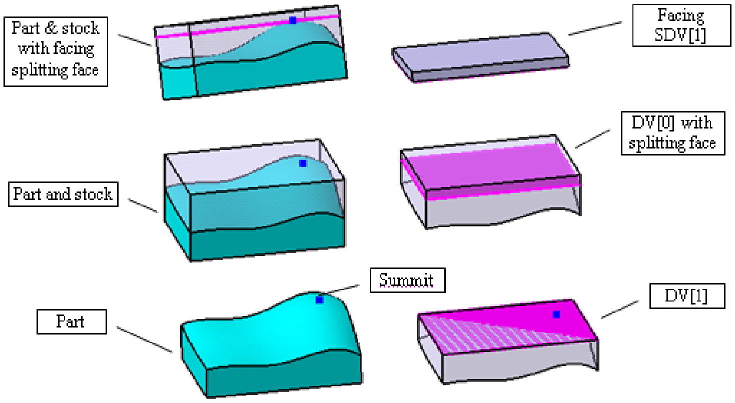

Part & stock with facing splitting face
Facing SDV[1]
DV[0] with splitting face
Part and stock
Summit
DV[1]
Part

Figure 3-3 A planar splitting face passing through the summit of a curved face

3.2.4 Creating a simple face

In the developed approaches, to separate a SDV from the DV, at least one face has to be created. As a created face does not belong to the boundary of DV[0], it is a virtual face passing the interior of DV, thus it does not leave any trace on the part. While watched from the space, a created face is completely embedded in a DV and it will be erased completely as the surrounding material is removed. Thus for the purpose of splitting there is no reason to create a face of two dimensional curvatures. So in this dissertation we do not discuss creating a splitting face with 3D curve edges. Planar face and face generated by sweeping a linear profile can be created with easy, furthermore they provide the benefit of facilitating the determination of tool access position and tool path in the downstream procedures, that why they are given the priories to be splitting faces. The issues about tool access position will be discussed in chapter 5.

3.2.5 Splitting faces

It is worthy to point out that in our approaches the face used to slice DVs can be an individual planar face, a face of one dimensional curvature, or the union of created faces connected by certain boundary faces of a DV. A splitting face (no a splitting plane) usually passes through at least one concave edge of the Brep component of a DV. A united splitting face is needed in cases that connected concave edges are not co-surfaced, or an inner loop exists in the face of a DV where a concave vertex is detected, as shown in Figure 3-4. When an inner loop is detected in a face *f* of DV, a face union passing one of the loop's edges can be used to split the current DV and get a SDV. To combine created faces into a face union at least one of the boundary faces of the current DV is needed.

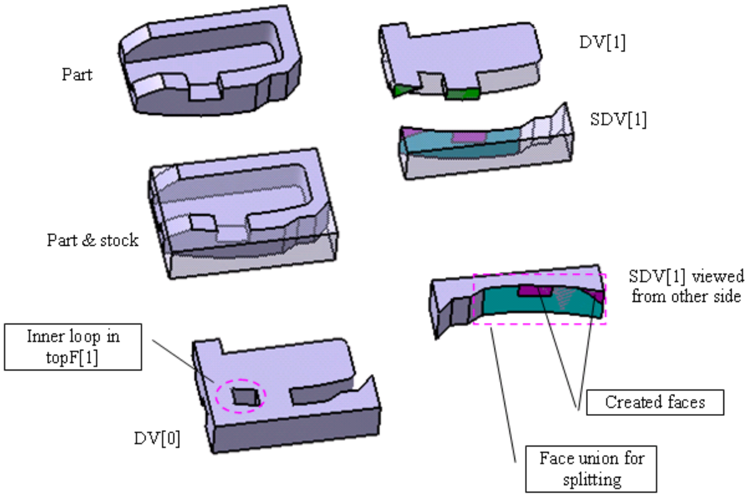


Figure 3-4 United splitting faces

3.3 Procedures and Algorithms

In the proposed approaches, decomposition is accomplished by cutting off a sub delta volume SDV[i] from delta volume DV[i-1], updating DV[i-1] to DV[i], then cutting another sub delta volume SDV[i+1] from DV[i], and so on, until no more concave edges can be detected in current delta volume. The delta volume is called a current DV if the newest SDV is going to be cut off from it. This procedure can be defined by a recursive equation:

$$\mathrm{DV}[i] = \mathrm{DV}[i\text{-}1] -^{*} \mathrm{SDV}[i], \ \mathrm{I} = 1, 2, \ldots \tag{3.3}$$

Brep model is employed for SDV representation, as well as for DVs, the part P and the stock S. The Brep of a SDV is determined in such way that a face, denoted spltF, separating the SDV from current DV is determined first, and then the SDV is consisted of spltF and all Brep elements of the current DV located on the specific (positive) side of spltF. In the following subsections the decomposition procedures and relative algorithms are presented.

3.3.1 Summary of decomposition approaches

The structure and primary procedures of the volumetric decomposition approaches developed in this thesis is illustrated in Figure 3-5. The Brep of a part P and a stock S are the input data, a set of SDVs is the output. Our approaches can be briefly summarized as follows, detail descriptions will be given later.

Step 1 Obtain the initial delta volume DV[0] by standardized Boolean subtraction of P from S. The geometry of DV[0] is represented in axis system XYZ that will be specified later in this section.

Step 2 Find out the face of DV[0] which is parallel to XY plane and passing the highest vertex of the stock. This face is recognized as the top face topF[0] of DV[0] which is used as the reference face for determination of first sub delta volume SDV[1], that is the facing sub delta volume.

Step 3 Determine the facing volume SDV[1] if it is applicable. Facing operation is needed only when top face topF[0] does not overlap with any geometry element of the part P, that is P $\cap$ topF[0] = $\varnothing$. In most cases facing is necessary, thus it is assumed that facing volume SDV[1] exists without exception.

Step 4 Update DV[0] to DV[i], I = 1, by determining the complement of SDV[1] in DV[0].

$$\mathrm{DV}[1] = \mathrm{DV}[0] -^{*} \mathrm{SDV}[1] \tag{3.4}$$

Step 5 Determine other external SDVs. An external SDV has a face being at least a portion of a stock boundary face.

Step 6 Determine interior SDVs. An interior SDV is enclosed by faces without overlap with any stock face.

In each step, when a new SDV is generated, the current DV will be updated to a new DV as defined in Equation (3.3). The decomposition process is terminated if no concave edge can be found in any boundary face of the current DV, and this DV will be taken as the last SDV.

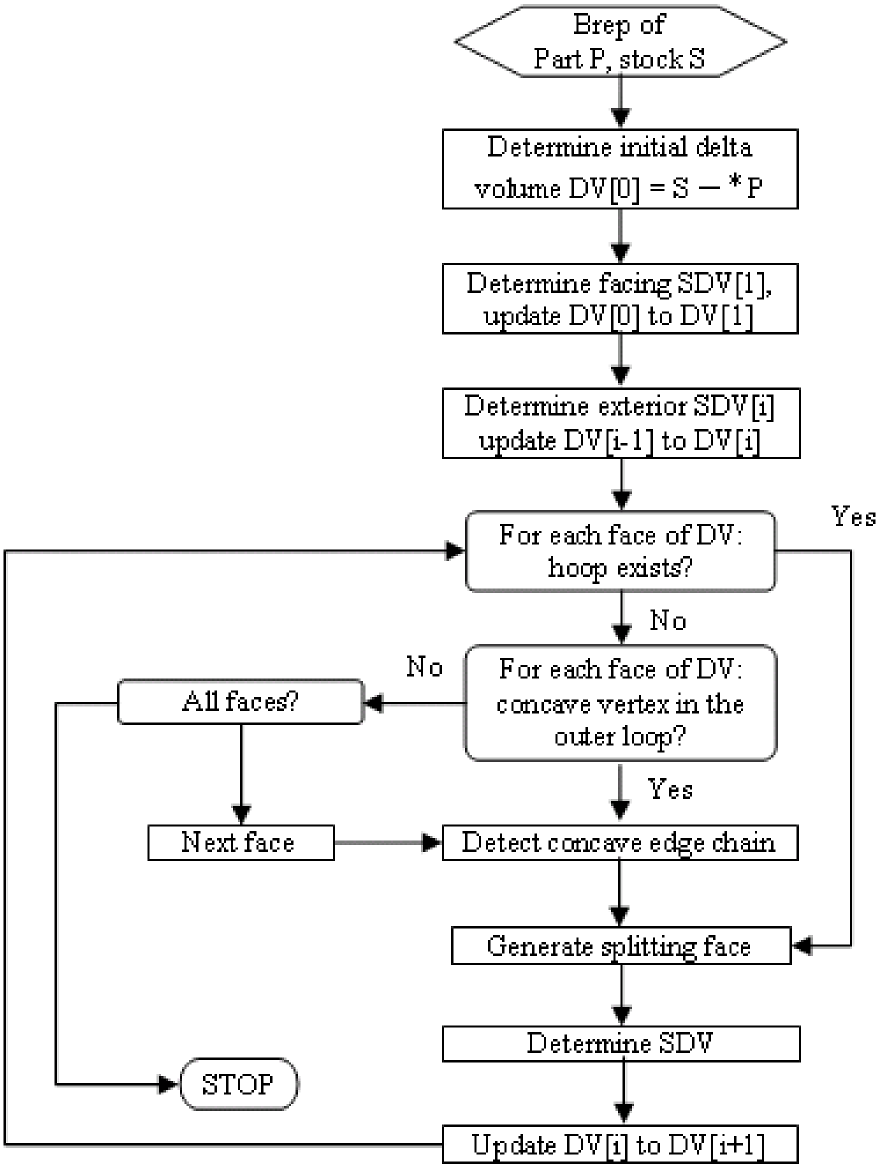


Figure 3-5 Flow chart of delta volume decomposition

3.3.2 Data structure

We assume that Brep topology organizes the data in a structure that can be represented in a vertex-edge-face table. The N individual vertices are stored in an array Vertex[i], I = 0, 2,…N-1, so they can be referred to by their indices in the array. Edges are represented by pairs of vertices indices. Faces are represented by boundary loops. A loop consists of an ordered list of vertex indices. The curve and surface geometries are associated with edges and faces respectively. In this dissertation, a curve is the mathematical representation of one-dimension geometry. An edge is a segment of a curve. A surface is the mathematical representation of a two-dimensional geometry. A face is a limited patch of a specific surface. The grammar of the table definition of [98] is adopted and expanded:

VertexIndex = 0 through N-1;

VertexIndexList = [VertexIndex V; VertexIndexList Vlist;]

EdgeList = [Edge E; EdgeList Elist;]

LoopList = [Loop L; LoopList Llist;]

FaceList = [Face F; FaceList Flist;]

Vertex = [VertexIndex V; EdgeList Elist; LoopList Llist; FaceList Flist;]

Edge = [VertexIndex V[2]; LoopList Llist; FaceList Flist;]

Loop = [VertexIndexList Vlist; EdgeList Elist;]

Face = [VertexIndexList Vlist; EdgeList Elist; LoopList Llist;]

The edge list Elist, loop list Llist and the face list Flist in the Vertex object are the lists of all edges, loops and faces respectively that have a vertex corresponding to the vertex indexed by V. The loop list Llist and face list Flist in the Edge object is the list of all loops and faces that share the specified edge. An edge element does not directly know if another edge shares one of its vertices. The loop list and face list in the Edge must be nonempty, because any edge in the collection must be at least a part of the loop of one face, otherwise the edge will be a dangle edge that does not belong to any volume. For the reason of not involving an isolated vertex, the face list and edge list in the Vertex must not be empty too.

In addition to the Brep data structure, we construct other two structures for representing functional classification of the boundary faces of a volumetric object. The faces of a volumetric object, a DV or a SDV, are classified into open face and closed face; and for the purpose of determining the tool access direction, we group the faces in three sets: top face, bottom face and wall face. We arrange the boundary faces of the volumetric object X as follows:

X.F = [OpenFList; ClosedFList; TopFList; WallFList; BottomFList;]

OpenFList = [OpenFace openF; OpenFaceList openFList;]

ClosedFList = [ClosedFace 51closedF; ClosedFaceList closedFList;]

TopFList = [TopFace topF; TopFaceList topFList;]

WallFList = [WallFace wallF; WallFaceList wallFList;]

BottomFList = [BottomFace 51bottomF; BottomFaceList bottomFList;]

Based on the face classification mentioned above, for description convenience, we further assume the following functions, as listed in , are available for extracting specific information from the Brep (topology and geometry) of an object.

In this thesis we do not distinguish a pointer from a duplicator of a geometric element. An assignment expression is used only to describe the equivalence of the term on the left side of the expression to the term, or the calculation result of the terms, on the right side of the expression. For example, the assignment openF[i] = F[k] means from then on openF[i] represents the same geometry as that possessed by F[k]. Data redundancy should be avoided by programming techniques when implementing the algorithms. For analysis purposes the same element of the Brep of an object can be indicated by variables of different names. For example, openF[i] and topF[j] may refer to the same face F[k].

3.3.3 Denotation of Initial data

The Brep models of a part P and a stock S are the primary input data for our approaches. The Brep of the initial delta volume DV[0] is obtained by standardized Boolean subtraction of P from S. We denote the Brep data in the follow structures:

P.face = [face F; face list Flist;]

P.edge = [edge E; edge list Elist]

P.vertex = [vertex V; vertex list Vlist;]

S.face = [face F; face list Flist;]

S.edge = [edge E; edge list Elist]

S.vertex = [vertex V; vertex list Vlist;]

DV[0].face = [face F; face list Flist;]

DV[0].edge = [edge E; edge list Elist]

DV[0].vertex = [vertex V; vertex list Vlist;]

Table 3-1 Functions for withdrawing specific information from Brep

Function Name	Description	Example
getface( )	Get the face list of an element	DV[0].Flist = getface(DV[0])
getloop( )	Get the loop boundary of a face	F.loop = getloop(F)
getedge( )	Get the edge list of an object	L.Elist = getedge(L)
getvertex( )	Get the vertex list of an object	SDV[1].Vlist = getvertex(SDV[1])
getsurface( )	Get surface (function) of a face	F.surface = getsurface(F)
getcurve( )	Get curve (function) of an edge	E.curve = getcurve $(E[i])$

Where X.face is the face list (or set of faces) of an object X, X.edge and X.vertex represent the edge list and vertex list of X respectively. Any topology element of an object can be cited by the index of the element in corresponding list. For example, as we have assumed that a rectangular block of stock is used, the faces of the stock can be indexed by S.face.F[i], I = 1 through 6. Data of other volumetric objects obtained by recursive decompositions, to be described in the following part of this chapter, are represented in the same way.

3.3.4 Coordinate systems

In many cases the coordinate system used by a design engineer to create the 3D model of a part differs from that used by a manufacturing engineer to determine the CAM program for producing the part. For example, the part is defined in the system xyz, Figure 3-6 (a), while the stock is defined in another system XYZ. One of the stock vertices is positioned at the origin and its three prism edges intersecting at this vertex are coincident with the coordinate axes respectively, as shown in Figure 3-6 (b).

To determine the initial delta volume DV[0], the part must be a subset of the stock volume as showed in Figure 3-6 (c). While taken XYZ as the working system, by coordinate system transformation if necessary, all geometry elements of the part as well as those of DV[0] are represented in the stock system XYZ, as illustrated in Figure 3-6 (d). The unit vector of axes X, Y and Z are denoted **u**, **v** and **w** respectively. It is assumed the coordinate system transformation is accomplished automatically for all geometry when a new reference system is chosen as the active system.

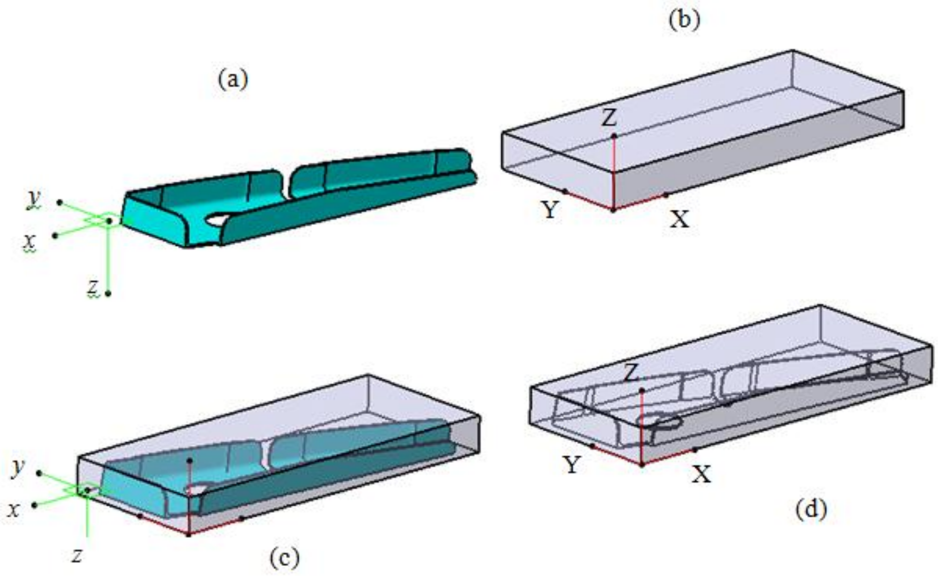


Figure 3-6 Reference systems of part, stock and DV

3.3.5 Facing volume SDV[1] determination

The facing volume SDV[1] is determined in three steps. The first step is taken to detect the top face; the second step determines the splitting face for separating SDV[1] from DV[0], the splitting face is considered as the bottom face of SDV[1]; and the third step concentrated on determining the wall faces that, together with the top face and the bottom face, will form the enclosure of SDV[1]. These three steps are explained in detail in the following part of this sub section.

3.3.5.1 Top face determination

In XYZ system, as defined above, the top face of DV[0], denoted DV[0].topF, can be defined as the top face of the stock, that is the face passing through the vertex of the maximal z coordinate and parallel to XY plane. The positive direction of a boundary face's normal is defined as pointing from the face outwards the volume. The top face of DV[0] is also the top face of SDV[1], and it can be determined by either of the following two algorithms:

Algorithm 3.1 – Top face determination 1

```
//Find out faces of S whose unit normal vector equal to w
For each face F∈S.face
    If F.normal n equals to w
    Then S.topF = F
End For
```

where **w** = (0, 0, 1) is the unit vector of axis Z while represented in the system XYZ.

Algorithm 3.2 – Top face determination 2

```
//Find out one of the highest vertex, denoted Vtop, from the vertex set of S by
    Vtop.z = max(V.z), V∈S.vertex
//Find out the face, denoted by S.topF, of S passing vertex Vtop and parallel to plane XY
    For each face F passing Vtop: F∈S.face ∧ F.vertex ⋂ Vtop = Vtop
        If F.normal = w
        Then S.topF = F
    End for
```

Either of the two algorithms above can be adopted for implement. The top face of S is also taken as the top face of SDV[1], as well as of DV[0], so

```
    SDV[1].topF = S.topF
```

Figure 3-7 illustrates the top face of DV[0] for example. For better view effect the origin is not show at one of the stock vertex.

3.3.5.2 Splitting face/Bottom face determination

As mentioned in section 3.2.2, passing through a vertex is not a necessity for a splitting face. Without loss of generality, we define the splitting face spltF[1] for separating facing sub delta volume SDV[1] from DV[0] as the face passing the maximum extremum of P in the direction **w**. The maximum extremum can be a point, a line segment or a planar face.

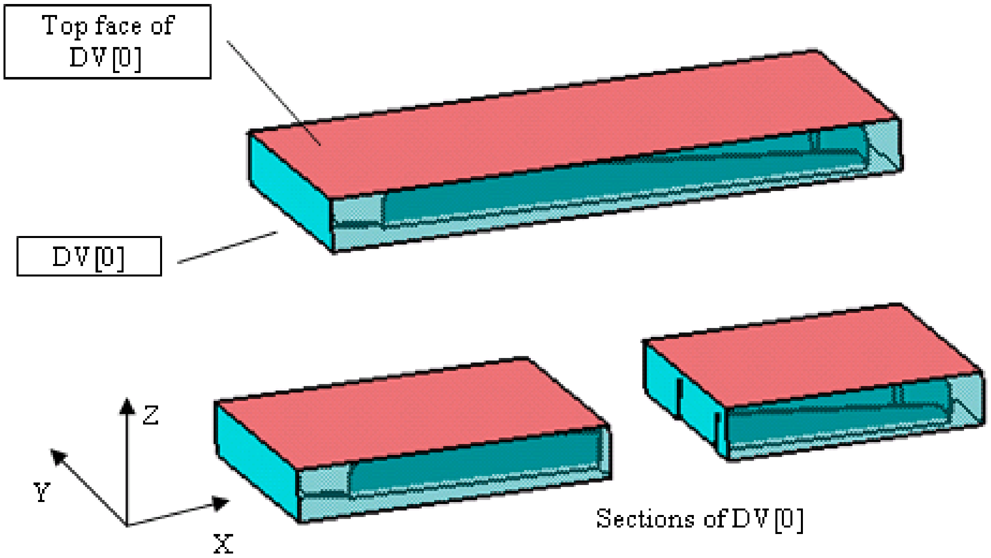


Figure 3-7 Top face of DV[0]

Some parts may have several discrete extrema of the same z coordinate in **w** direction. These extrema can be the same or different geometric categories. As the splitting face overlapping with any one of these elements will overlap all of them, thus without loss generality, we can choose the splitting face passing the extremum element closest to the reference system plane YZ.

Denote $\mathrm{getextrem}\left(\mathrm{P},\mathbf{w}\right)$ for detecting the extremum of an object P along a specified direction **w**. The outputs of $\mathrm{getextrem}\left(\mathrm{P},\mathbf{w}\right)$ are organized in the type of planar face, line segment and point, denoted by extrem( ).F, extrem( ).E, extrem( ).V respectively. Denote $\mathrm{intersect}\left(x,\ y\right)$ the function detecting and output the intersection of geometries elements *x* and *y*.

First, determine the surface, denoted spltSurf[1], of which the splitting face is a bounded portion. Then determine the splitting face spltF[1] by finding out the intersection of the splitting surface spltSurf[1] and the initial delta volume DV[0]. Algorithm 3.3 is created to determine the splitting face for separating SDV[1].

Algorithm 3.3 – Splitting face spltF[1] determination

```
        $\mathrm{getextrem}\left(\mathrm{P},\mathbf{w}\right)$
        If $\mathrm{extrem}\left(\mathrm{P},\mathbf{w}\right).\mathrm{F}\neq\varnothing$
//extrem(P, w).F is planar
                spltSurf[1] = getsurface(extrem(P, w).F)
                else if $\mathrm{extrem}\left(\mathrm{P},\mathbf{w}\right).\mathrm{E}\neq\varnothing$
                        spltSurf[1] = getplane(extrem (P, w).E.V[0], w)
                else if $\mathrm{extrem}\left(\mathrm{P},\mathbf{w}\right).\mathrm{V}\neq\varnothing$
                        spltSurf[1] = getplane(extrem (P, w).V, w)
                End if
                spltF[1] = intersect(spltSurf[1], DV[0])
        End of Algorithm 3.3
```

The bottom face of SDV[1] is determined by the following assignment

$$\mathrm{SDV[1].bottomF = spltF[1]}$$

Figure 3-8 shows the position of detected extremum, splitting surface spltSurf[1] and the splitting face spltF[1].

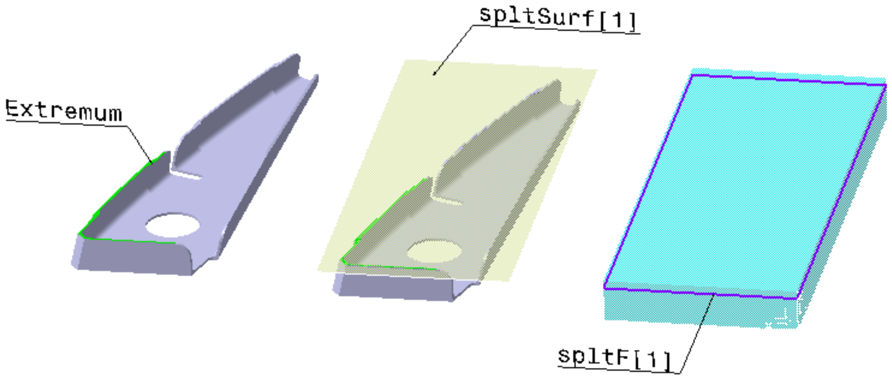


Figure 3-8 Extremum and splitting face

3.3.5.3 Volumetric SDV[1] determination

The volumetric SDV[1] can be determined in two ways. One of them is to determine the portion of DV[0] in the positive half space of the splitting surface spltSurf[1], and the other method is to split DV[0] with the specified face spltF[1]. Both methods result the same sub delta volume SDV[1] in this case. But in some cases splitting with a surface conducts results different from that with a face, because a surface is unbounded thus it may intersect a delta volume at several intersection faces, and most often some of these intersections are not the places where the splitting is intended to happen. On the other hand splitting can be well controlled by using a specified face. The volume should be sliced only at specified locations. So instead of using a surface, we prefer using a splitting face to separate a SDV from its father DV.

Denote split(B, F, **t**) the function for splitting volume B with splitting face F. The output of this function is the portion of B situating on the side of F indicated by **t**. If F is a union of faces, **t** is a set of indicators that constrains the topologic elements consisting of the Brep of SDV. The principle of the function split(B, F, **t**) can be described as follows:

1) Find out vertices Vs of the Brep of B located on the positive side of F indicated by **t**;

2) Determine the intersection of F with faces passing Vs;

3) Trim the intersected faces with F and keep their portions on the positive side of F;

4) The resultant volume is bounded by F, the trimmed faces, and face(s) passing any one of Vs but not trimmed by F.

The key issue of splitting is the determination of the side indicator **t**. For spltF[1], the side indicator **t** is determined by $\mathbf{t} = \mathbf{w}$, thus the portion of DV[0] on the positive side of spltF[1] constitutes SDV[1]. The side indicators of other situations will be determined in algorithms presented later in this section. Figure 3-9 shows an example of determining the volumetric SDV[1].

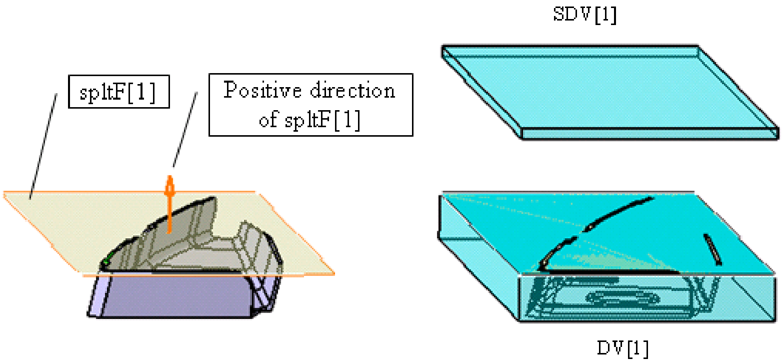


Figure 3-9 Determination of volumetric SDV[1]

<u>Algorithm 3.4</u> – Volumetric SDV[1] determination

```
     given splitting face spltF[1]
     //Let the positive normal t of spltF[1] be equal to w
     $\mathbf{t} = \mathbf{w}$
   //Determine SDV[1] by assigning it the result of function split( ) with DV[0],
   //spltF[1] and t as the inputs
           SDV[1] = split(DV[0], spltF[1], t)
End
```

3.3.5.4 Wall face determination

Excluding the top face and bottom face, the rest enclosure faces of SDV[1] are wall faces of the sub delta volume. As SDV[1] is a 3D manifold determined by Algorithm 3.4, the list of its enclosure faces SDV[1].face can be obtained by applying the function getface( ). The list of wall faces is the complement of the top face and the bottom face to the set SDV[1].face, see Figure 3-10. Algorithm 3.5 is developed to find out the wall faces of SDV[1].

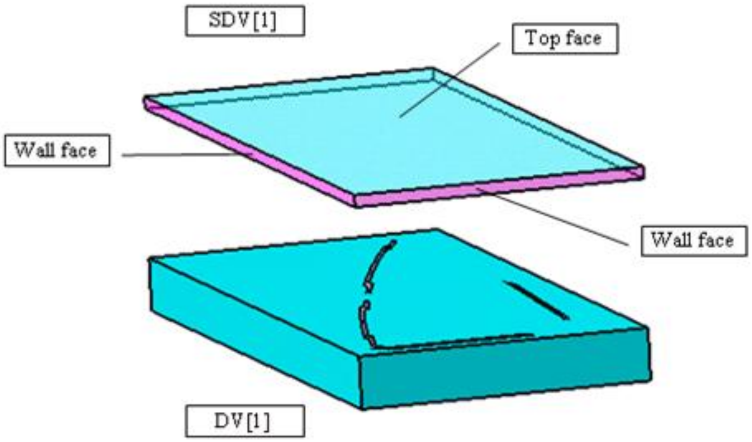


Figure 3-10 Wall faces of SDV[1]

Algorithm 3.5 – Determine wall faces of SDV[1]

```
//get the enclosure faces of SDV[1] and keep them in SDV[1].face
        SDV[1].face = getface(SDV[1])
//initial index number of wall faces
        k = 0
        For each face F[j] $\in$ SDV[1].face
        //if F[j] equals neither to the top face nor to the bottom face of SDV[1]
        //then F[j] is a wall face
                If F[j] $\neq$ SDV[1].bottomF $\wedge$ F[j] $\neq$ SDV[1].topF
                    SDV[1].wallF[k] = F[j]
                        k increments
                end if
        End for
```

Algorithm 3.5 can be written in a simpler manner,

```
        SDV[1].face = getface(SDV[1])
```

SDV[1].wallF = SDV[1].face – (SDV[1].topF + SDV[1].bottomF)

3.3.5.5 Determination of openF and closedF

In order to know whether a face of a specified SDV is accessible by a tool approaching from the air without cutting other SDVs, we classify the faces of a SDV into two categories: open face and closed face.

An open face is a face exposed to the air if all adjacent SDV are omitted. An open face permits a tool to pass through during machining.

A closed face is the face overlapping with a part face, which should be created by machining. A tool never passes through a close face but moves along a curve in the face, meanwhile staying on the negative side of the face.

At this stage we define a face as an open face if it is at least a portion of a stock boundary face or it is created by splitting operation and the splitting face does not interest the part. Other faces enclosing the part are s. Algorithm 3.6 is developed to determine the open face and closed face of SDV[1].

Algorithm 3.6 – openF and closedF determination

```
S.face = getface(S)
SDV[1].face = getface(SDV[1])
For each face F[j] $\in$ SDV[1].face
    If F[j] $\cap$ S.face = F[j]
        SDV[1].openF[k] = F[j]
    else
        if F[j] = spltF[1] and intersect(spltF[1], P) = $\varnothing$
            SDV[1].openF[k] = F[j]
      else F[j] is a
            SDV[1].closedF[n] = F[j]
end
```

3.3.6 Determination of exterior SDVs

An exterior SDV is a portion of DV[1] whose enclosure contains at least a portion of a stock face. Exterior SDVs are portions of the removable volume of material that surround the outwards pointing faces of a part. The main procedures for exterior SDV determination are shown in Figure 3-11.

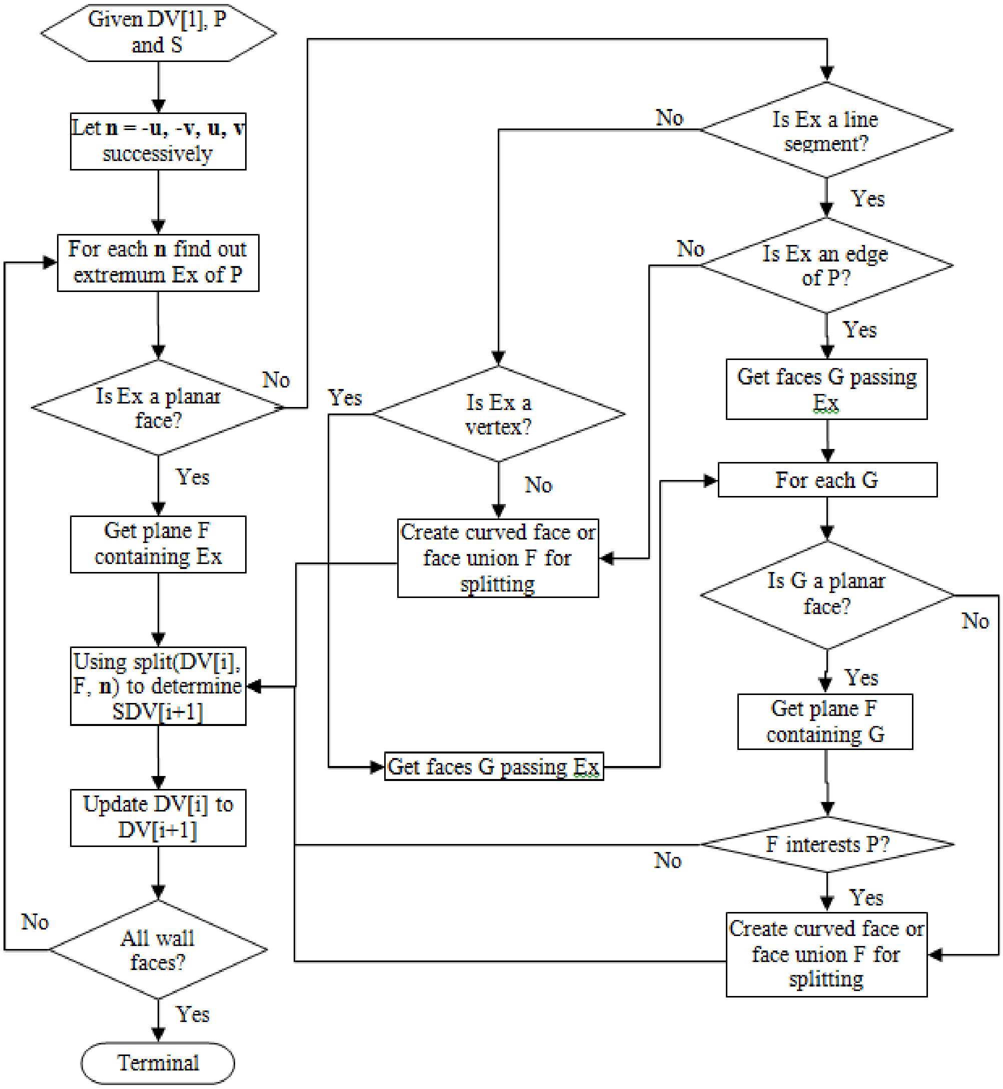


Figure 3-11 Flowing chart for exterior SDV determination

To determine the exterior SDVs, we adopt the approach of detecting the extremum of a volume in a specified direction as that has been applied to the determination of top facing volume. But this time more situations have be considered. First, we have to consider the possibilities of exterior

SDV existing on four sides of the part. Thus function bottom62(P, **n**) needs to be applied to part P four times, each time **n** will be assigned one of the four direction vectors $\{\mathbf{u}, \mathbf{v}, -\mathbf{u}, -\mathbf{v}\}$ respectively. Second, the extremum can be a planar face, a line segment or a point. Different algorithms are developed for creating splitting faces corresponding to individual types of extrema.

3.3.6.1 Splitting with a single face

The proposed approaches are developed according to the geometry of the extremum Ex obtained: 1) a planar face, 2) a linear segment, 3) a point. In case 2) and 3) situations are considered separately: Ex is or is not a boundary element of the part.

Case 1 The detected extremum Ex is a planar face as shown in Figure 3-12 (b). It is obvious that face Ex is parallel to the stock face possessing normal **n**. Determine plane G = getsurface(Ex), then G can be used to split current DV. Under this circumstance, splitting current DV with either G or the intersection of G with the DV will produce the same result. The positive side of G is also indicated by **n**.

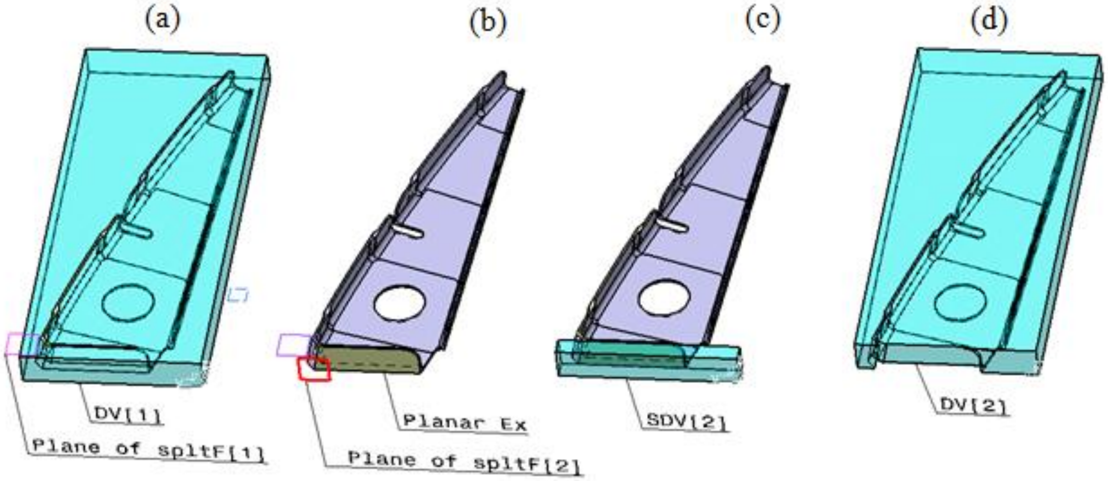


Figure 3-12 Planar face Ex and SDV[2]

Case 2 Ex is a line segment. Under this circumstance two situations have to be further considered: 1) Ex is an edge of the part's BRep, which indicates the intersection of two part faces, and 2) Ex is not an edge element of the part's Brep, but just the absolute "maximum" of a part face in direction **n**. In both situations, the segment Ex is actually perpendicular to **n**.

Case 2.1 In the case that Ex is an edge of the part, as having discussed in 4.3.2, apply function getface( ) to the line segment Ex to get the faces sharing edge Ex. Denote $\{f,b\}$ the two faces sharing Ex, $\{f,b\} = \text{getface(Ex)}$. Suppose face *f* does not overlap with any created splitting face, and it situates on the left or on the top of *b* as viewed in the direction –**n**, the splitting face to be created depends on *f*. Otherwise face *b* will be taken into account. More situations are taken into account as follows:

C2.1.1 If the face *f* is planar, and if the plane G containing *f* does not intersect with the part P, then the intersection of G with current DV will be taken as the splitting face used to separate the exterior SDV from current DV.

C2.1.2 If G intersects with P, we can simply create a plane *h* passing Ex and perpendicular to direction **n**. It is obvious that plane *h* does not intersect with the part. Thus the intersection of *h* and the DV can be one of the candidates of the splitting faces. From the perspective of machining efficiency, the volume of removable material should be apportioned as much as possible to exterior sub delta volumes. We know that, from both theoretical and practical aspects, there is no need for a tool to pass the whole volumetric space of exterior SDVs to produce the outside shape of a part. Thus if G intersects the part, and if all faces $\{\beta\}$ of the part that intersected by G are not adjacent to face *f*, we can determine the extrema of the loops enclosing $\{\beta\}$ in the same direction **n**, and the one having the maximum distance to the system reference plane perpendicular to **n** is taken as the second extremum ex2. Then a plane passing Ex and ex2 is generated for splitting. If there is a face $\varphi \in \{\beta\}$ adjacent to *f*, a face union is suggested. Creation of face union will be discussed in section 3.3.6.1.

C2.1.3 Face *f* is a curved face, the surface *s* of *f* will first be verified, if it does not intersect the part before intersecting the stock faces, the intersection of *s* with the DV is then the splitting face. Otherwise, if *s* intersects the part before intersecting the stock faces, and if all faces $\{\beta\}$ of the part intersected by *s* are not adjacent to face *f*, we can create a plane in the way as mentioned in C2.1.2 to determine the splitting face.

Case 2.2 Ex is not an edge of the part. In this case we have to detect the part face that has Ex as its extremum in direction **n**. The part face can be found out in two steps. Step 1, find out the edges e1 and e2 intersecting with Ex at the two end points of Ex. Step 2, find out the loop consisted of both e1 and e2, thus determine the face *f* passing Ex. Then a spitting face can be generated in the same way mentioned in situation 2.1.3.

Case 3 The detected extremum is a point Q, it can be a vertex of the part, a point in the interior of an part edge, or a summit of a curved part face in the specified direction. In the first situation we can find out the face *f* from all the faces passing Q in such a way that the normal of face *f* at Q forms the smallest angle with vector **n**, and create a splitting face according to *f* in one of the ways discussed above. In the other two situations, we have to detect the part edge *e* and the part face *f* passing Q respectively, hereafter to create the splitting face according the face *f* as discussed above.

In those cases discussed above, it is supposed that the plane *g* or surface *s* dose not intersects with a part face adjacent to *f*. If the intersected face is adjacent to *f*, splitting with a single face will lead to leave more material to interior SDVs. To remove an interior SDV a tool have to sweep the entire volume of the SDV. In order to generate SDVs of machining efficiency, using a union of faces, created according to the outside shape of the part, for splitting is preferred. The approach to creating a face union for splitting will be discussed in the following subsection.

A plane used to generate a splitting face must not intersect with the part. Thus excepting the plane containing a planar face extremum, which is inherently free of intersecting the part, all other splitting planes have to be evaluated before they are applied to slice the DV. Figure 3-12 illustrates an example of exterior SDV determination when the detected extremum is a planar face.

The given DV[1] and correspondent stock axis system XYZ are shown in Figure 3-12 (a). The face *f* of the part P, with a normal of $\mathbf{n} = -\mathbf{u}$, is detected as a planar face extremum, thus the plane containing of *f* is taken as the splitting plane, its intersection with the stock is the splitting face denoted spltF[2], as shown in Figure.12(b). The exterior sub delta volume SDV[2] is determined by applying function split( ) taking DV[1] (the current DV), spltF[2] and **n** as the input parameters

$$\mathrm{SDV}\left[2\right] = \mathrm{split}\left(\mathrm{DV}\left[1\right],\ \mathrm{spltF}\left[2\right],\mathbf{n}\right) \tag{3.5}$$

then DV[1] is renewed as DV[2] by Boolean subtraction of SDV[2] from DV[1],

$$\mathrm{DV}\left[2\right] = \mathrm{DV}\left[1\right]-\mathrm{SDV}\left[2\right] \tag{3.6}$$

shown in Figure 3-12 (c) and (d). Other exterior SDVs of the same part, which are also cut off by plane faces, are illustrated in Figure 3-13.

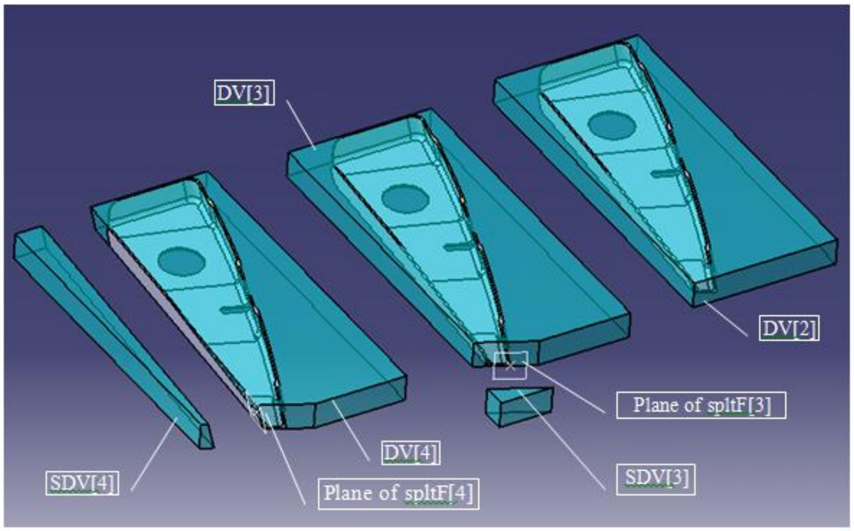


Figure 3-13 Exterior SDVs split with planes

3.3.6.2 Create face union for splitting

As mentioned above, in the case that the plane *g* or surface *s* intersects the part at a face which is adjacent to *f*, a face union needs to be created to conduct a splitting.

Naturally none of the DV's boundary face can play the role of splitting face independently to separate a portion of volume from current DV, because for being a splitting face at least a portion of the face must be embedded in the DV. The existing boundary faces of a DV does not meet this criteria as none of them has any intersection with the interior of the DV, they just bound the DV.

In order to generate a union splitting face correspond to the part shape, surfaces of related part faces have to be employed. In fact, the face union is the combination of portions of the intersections of these surfaces with current DV. The key task is to choose part faces whose surfaces will be used in determining the face pieces of the union. Certain prerequisites are considered for choosing the face, as listed below.

1) All faces to be selected have to be adjacent one to the other at their column edges. A column edge of a face is an edge that has a direction corresponding to Z-axis. To determine the column edges of a face *f*, we verify the angles between the unit vector **w** of the Z-axis and the tangents **t** of individual edges enclosing *f* at their "to" vertices respectively, the two edges of the greatest and smallest angles are considered as the two column edges of the face. These faces form a connected face series including the face *f* that contains the extremum of the part P in the direction **n**, as illustrated in Figure 3-14, the first selected face is denoted f[1].
2) Faces passing extrema of the part in other directions and not share the present Ex should not be included.
3) Faces fully overlapping with any previous splitting faces are not included.

Gap filling: In order to generate a face union without including the volume of a slot in the exterior SDV, faces of a slot should not be involved in the list of selected faces. A face filling the gap has to be created, see Figure 3-14. The task of generating face pieces, which are components of the face union, is accomplished by the following steps.

Step 1 Determine the extrema {Ex} of the part P in directions $\mathbf{n} \in \{\mathbf{u}, \mathbf{v}, -\mathbf{u}, -\mathbf{v}\}$;

Step 2 for each direction **n,** except those having corresponding splitting faces already, identify face f[1] of the part P passing Ex corresponding to **n** and not overlapped with any previous splitting face; if two faces share Ex, the one on the right as viewed in direction –**n** is selected as f[1]; if more than two faces share Ex, then Ex is one of the vertices of P. Denote Ex.Elist the list of edges sharing Ex. Let $\{\mathbf{e}_i\} = \text{Ex.Elist}$, if

$$-\frac{\mathbf{e}_k}{\|\mathbf{e}_k\|}\mathbf{w} = \max\left(\frac{\mathbf{e}_i}{\|\mathbf{e}_i\|}\mathbf{w}\right),\ \mathbf{e}_k, \mathbf{e}_i \in \{\mathbf{e}_i\} \qquad (3.7)$$

and the face *f* consisted of edge $\mathbf{e}_k$ is not totally overlapped with any previously created splitting face, then ***f*** is selected as f[1]. Equation (3.7) implies that among all edges sharing Ex, $\mathbf{e}_k$ forms the smallest angle with –**w**, thus the face *f* consisted of $\mathbf{e}_k$ located on the right side of $\mathbf{e}_k$ as viewed from –**n** while stand up with head in direction **w**.

Step 3 Determine column edges e1 and e2 of f[1];

Step 4 For each column edge find out the face g adjacent to f[1], g is not totally overlapped with any previously defined splitting face. Without loss of generality, we discuss only the situation that g is on the left side of f[1] as viewed in the direction opposite to the normal of f[1]. For expression convenience face g is denoted f[2].

Step 5 Repeat Step 3 and Step 4 on face f[2], and so on to get a series of faces denoted selectedFlist. To avoid including both inward and outward faces of a thin wall of the part in the series, it is required that the column edges of f[i] used to determine f[i-1] and f[i+1] have to be connected by the lowest edge of f[i].

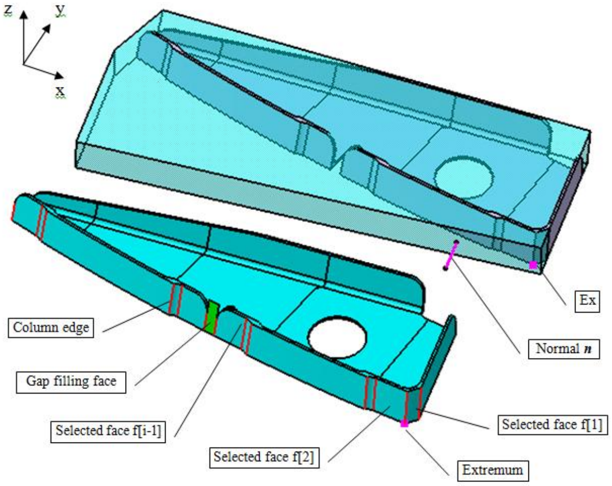


Figure 3-14 Extremum, Column edge and gap filling face

Step 6 Detect a gap in order to make an exterior SDV without slots. This research is limited to typical form of slot in aerospace structural part, which consists of two parallel faces with opposite normal vectors. According to this characteristic a slot and then faces forming the slot can be identified. Eliminating slot faces from the face series makes a gap in the connected faces.

Denote selectedFlist = $\left\{ f_j \right\}$ the set of selected faces. Denote $\left\{ \mathbf{n}_j \right\}$ the set of normal vectors at the centers of faces in $\left\{ f_j \right\}$. If $\exists \mathbf{n}_i, \mathbf{n}_k \in \left\{ \mathbf{n}_j \right\}$

$$\mathbf{n}_i\mathbf{n}_k = -1,\ i < k \tag{3.8}$$

then face f_i and f_k are the start and end faces of a slot. Find start and end faces of all slots in $\{f_j\}$. Denote slotDetect = {sd$_i$} the set of slots, sd$_m$ = $\{f_i, f_k\}$is the couple of start and end faces of slot sd$_m$. Remove f_i , f_k and faces listed between them in $\{f_j\}$, thus make a gap *gp$_m$*.

Step 7 Create a face *ff$_m$* filling the gap *gp$_m$*. The face is determined by a loop of four boundary edges. Two of them, E1 and E3, are the column edges at the beginning of the gap, or the extended column edges limited by the top and the bottom of the stock (or current DV). The other two edges of the loop, E2 and E4, are line segments joining E1 and E3 at their end points. Edge E1 is the segment of the intersection of the surfaces of f_i and f_{i-1} limited by stock faces. Similarly edge E3 can also be determined.

Step 8 Rearrange the face elements of selectedFlist, by replacing f_i by *ff$_m$* , and update the index of the faces succeeding f_i . Denote $\{g_j\}$ the updated list of selected faces

$$\begin{cases} g_j = f_j,\ j < i \\ g_j = ff_m,\ j = i \quad ,\ f_j \in \{f_j\} \\ g_{j-(k-i)} = f_j,\ j > k \end{cases} \tag{3.9}$$

Equation (3.9) is applicable to the situation that only one slot is detected in$\{f_j\}$. To save space formula for updating the list of faces after removing multiple slots is omitted.

Step 9 Get surface of each face *g* in $\{g_j\}$ if the column edges of *g* do not ended at spltF[1] and the bottom face of the stock, denoted S.bottomF, and arrange these surfaces in a surface list denoted selectedSlist. If vertices of both column edges of *g* are in spltF[1] and S.bottomF, then face *g* is one of the face pieces to be united.

Step 10 For each surface *s* $\in$ selectedSlist, if both column edges of the selected face *g* corresponding to *s* has a vertex on the stock bottom, a face piece is generated by bounding the intersection of *s* and the current DV with the following geometries: spltF[1], S.bottomF and the surfaces of the faces adjacent to *f* at its column edges. Otherwise, denote e1 and e3 the column edges of *f*, suppose the direction of e3 is opposite to **w**. instead of using S.bottomF, use the surface(s) of the face(s) adjacent to *f* at the edge(s) between e1 and e3 commencing from e3.

Put all face pieces in a list denoted pieceFList = $\{pieceF[i]\}$ $(i=1,2,\cdots \mathrm{N})$ in accordance with the order of their corresponding faces listed in $\{g_j\}$.

Step 11 Combine all faces in $\{pieceF[i]\}$ into a face union denoted UF.

$$\mathrm{UF} = \bigcup_{i=1}^{N} pieceF[i] \tag{3.10}$$

Step 12 The last step is to determine the positive side of UF. According to the previous procedures taken to form UF, it is obvious that UF is not a Möbius strip. One of the properties of Möbius strip is side continuously turned over. Let a vector crossing a Möbius strip move along the center line, meanwhile keeping its ends on the two side edges of the strip respectively, it will turn reverse having traveled the entire length of the strip. A Möbius strip is closed band. In most cases a UF is open at two ends. In case a closed UF is yield, it cannot be a Möbius strip because its edge on the top is consisted of curves in the same plane, the plane of spltF[1] if a facing SDV is separated first, and points on any other edges of UF are below the plane. A vector bridging two edges of UF never turns reverse no matter how it travels in UF, keeping its head on the top edge of UF.

As UF consists of multiple faces with different normal vectors, even with infinitive normals in the case of a curved face, it seems impossible to have a unique direction to determine the positive side of UF. Thus the positive sides of individual faces in the union are used for conducting the splitting. We define the positive side of face F in the way that given the normal vector **m** at the geometric center of the face, see Figure 3-15, if the angle between **m** and **n** is not greater than $\frac{\pi}{2}$ **,** then **m** defines the positive side of F. Otherwise the direction corresponding to normal of the closest stock face is the positive. The normal vectors of individual faces of UF are organized in set UF.normal.

3.3.6.3 Splitting with UF

To split a DV[i] with a face union UF is similar to that with a single face. As the UF is consisted of faces adjacent one to the other and all faces in UF are extended to intersect with both the top face and the bottom face, UF separates the DV in two parts situating on each of its sides respectively. With the determination of the positive side of UF, we assume that the splitting function split( ) is also applicable to face union and the set of its normal vectors. Thus the exterior SDV[i] and resulted DV[i+1] can be determined by the following algorithm.

Algorithm 3.8 – Splitting with UF

```
    Given DV[i] to be split
    Given face union UF and the set of normal vectors UF.normal
    SDV[i] = split(DV[i], UF, UF.normal)
    DV[i+1] = DV[i] $-^{*}$ SDV[i]
End
```

The exterior SDV determining procedure is terminated when the approaches have been conducted to all four directions, ± x and ± y.

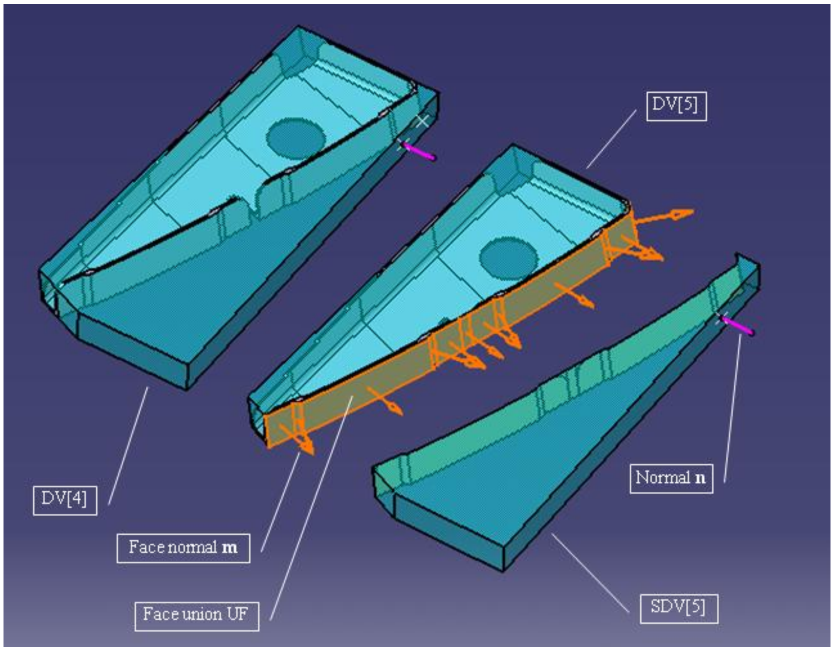


Figure 3-15 Splitting with face union UF

3.3.7 Basic Concepts for identifying interior SDV

Before presenting algorithms of separating island and other interior SDVs from related DV, some concepts are worth to be introduced.

Orientation of loop and edge

For faces in a DV, suppose the vertices of an outer loop L bounding a face F are clockwise cyclically ordered as viewed opposite to the local normal vector **n** of F, see Figure 3-16. An inner loop is oriented counter-clockwise. The normal **n** directs toward the exterior of the volume bounded by the face. The direction of an edge in a loop is determined by their ordered vertices. According to this definition, the interior of a face is always on the right side of the edge referenced by it; vertices bounding a face of DV are listed in the same sequence as corresponding vertices bounding a part face; the normal of a DV face is in the direction opposite to that of a par face at corresponding point.

Concave vertex

Let the tangent vector **t** of an edge be in accordance with the direction of the edge. Assume in the outer loop L of face F, edges E1 and E2 are intersect at vertex *p*. Let $\mathbf{t}_1$ and $\mathbf{t}_2$ denote the edge tangent vectors of E1 and E2 at their common vertex *p*, shown in Figure 3-16.

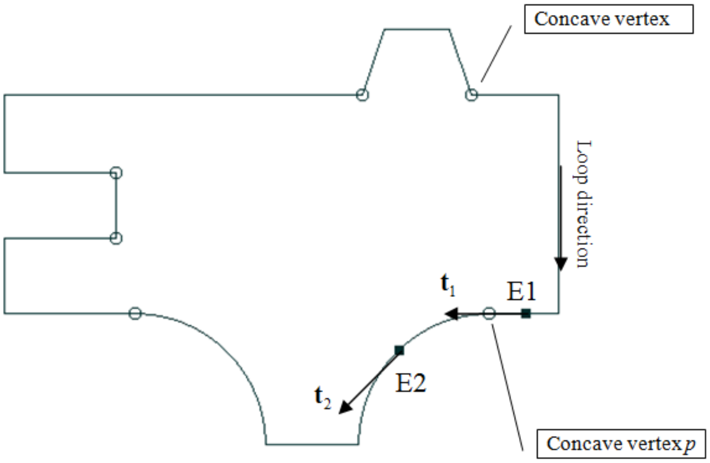


Figure 3-16 Example of concave vertices

Suppose *p* is the "to" vertex of E1 and the "from" vertex of E2. Vertex *p* is a concave vertex if the following inequality is satisfied

$$(\mathbf{t}_1\times\mathbf{t}_2)\mathbf{n}>0,\ \mathbf{t}_1\times\mathbf{t}_2\neq 0 \tag{3.11}$$

Where **n** is the normal vector of F and pint p.

If $\mathbf{t}_1\times\mathbf{t}_2=0$, find point q on E1 by

$$q=\text{extrem}\left(\text{E2},\ \mathbf{r}\right),\ \mathbf{r}=\mathbf{n}\times\mathbf{t}_1 \tag{3.12}$$

Replace $\mathbf{t}_2$ by vector $\overrightarrow{pq}$ in inequality (3.11), if the inequality is satisfied, vertex p is a concave vertex. Otherwise it is convex. In Figure 3-16, for better view effects vectors $\mathbf{t}_1$ and $\mathbf{t}_2$ are shown at the middle points of E1 and E2 respectively.

Concave edge

Suppose faces F_1 and F_2 are adjacent at edge E. Passing an arbitrary point p on E, make plane PL perpendicular to E. The intersections of PL with F_1 and F_2 are denoted by C1, C2 respectively, see Figure 3-17.

$$\begin{cases}\text{C1}=\text{F}_1\bigcap\text{PL}\\ \text{C2}=\text{F}_2\bigcap\text{PL}\end{cases} \tag{3.13}$$

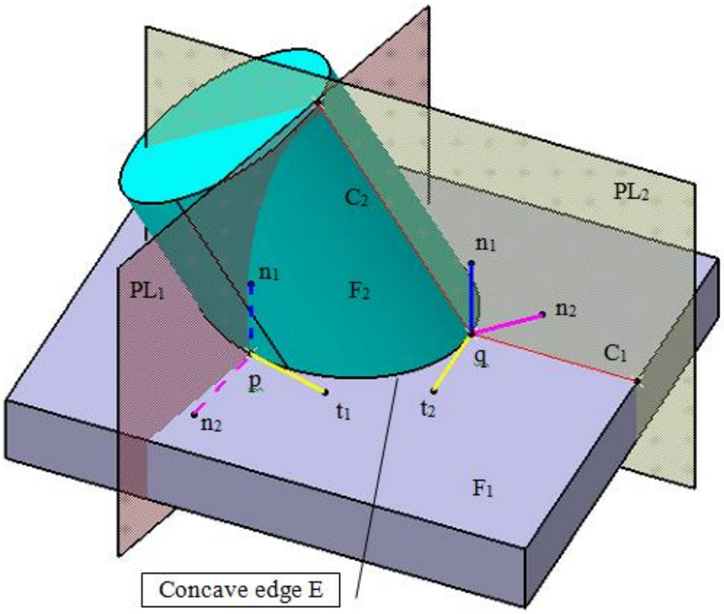


Figure 3-17 Example of concave edge

Let $\mathbf{n}_1$ and $\mathbf{n}_2$ denote the normal vectors of F_1 and F_2 at *p*. If C1 and C2 are tangent at *p*, $\mathbf{n}_1$ and $\mathbf{n}_2$ represent the normal vectors at the middle points of C1 and C2 respectively. Let $\mathbf{t}_1$ denote the tangent vector of E at point *p* referenced by F_1, and $\mathbf{t}_2$ denote the tangent vector of E at point *p* referenced by F_2. The edge E is concave if

$$(\mathbf{n}_1 \times \mathbf{n}_2)\mathbf{t}_1 > 0 \tag{3.14}$$

Or

$$(\mathbf{n}_2 \times \mathbf{n}_1)\mathbf{t}_2 > 0 \tag{3.15}$$

Otherwise

$$(\mathbf{n}_1 \times \mathbf{n}_2)\mathbf{t}_1 < 0 \tag{3.16}$$

$$(\mathbf{n}_2 \times \mathbf{n}_1)\mathbf{t}_2 < 0 \tag{3.17}$$

Figure 3-17 illustrates an example of concave edge while C_1 and C_2 are not tangent to each other. Edge E is the intersection of face F_1 and F_2. Plane PL_1 and PL_2 pass points p and q respectively. Both of them are perpendicular to edge E. Vectors $\mathbf{n}_1$ and $\mathbf{n}_2$ are normal vectors of faces F_1 and F_2 respectively (shown at both point p and q). Tangent vector $\mathbf{t}_1$ of E is oriented as it is referenced by F_1, while $\mathbf{t}_2$ is referenced by F_2. It is intuitive that both $\mathbf{n}_1$ and $\mathbf{n}_2$ are in plane PL_1 (or PL_2), the cross product $\mathbf{n}_1 \times \mathbf{n}_2$ is in the same direction of $\mathbf{t}_1$ when E is concave, while $\mathbf{n}_2 \times \mathbf{n}_1$ points to the direction of $\mathbf{t}_2$.

In Figure 3-17 edge E is an inner loop of face F_1. As assumed in this research the interior of a face is on the right side of any loop boundary the face, the positive direction of an inner loop is counter-clockwise as viewed opposite to the local normal of the face.

Left/right surface

Given an oriented edge, the surface consisting of the face on the left/right side of the edge is defined as its left/right surface, denoted LS/RS respectively.

Front/back concave vertex

For a concave vertex CV in the outer loop of face F, find out the closet side face sF of the stock. Denote $\mathbf{n}$ the normal of sF, suppose concave edge *e* has vertex CV as its “from” vertex, denote LF and RF the left and right faces of *e* respectively, LS and RS the surfaces of LF and RF respectively. Denote **n**LF and **n**RF the normal vectors of LF and RF at CV respectively. Vertex CV is a front concave vertex (front vertex simply) if

$$\mathbf{n}\mathrm{LF}\cdot\mathbf{n} > \mathbf{n}\mathrm{RF}\cdot\mathbf{n} \qquad (3.18)$$

otherwise CV is a back concave vertex (or back vertex), as illustrated in Figure 3-18.

If CV has equal distances to two or three stock faces, the stock face sF whose normal **m** has the direction corresponding to **n**LF will be selected, and the term **n** in criterion (3.18) will be replaced by **m**.

Able to be trimmed

Given faces F_1 and F_2 of current DV, denote S_1 and S_2 the surfaces of F_1 and F_2 respectively. Suppose S_1 and S_2 intersect each other

$$C = S_1 \cap S_2 \neq \varnothing \qquad (3.19)$$

Surfaces S_1 and S_2 are able to be trimmed each other if

$$S_1 \cap S_2 \cap DV \neq \varnothing \qquad (3.20)$$

A simpler way to verify the possibility of being trimmed can be expressed as follows.

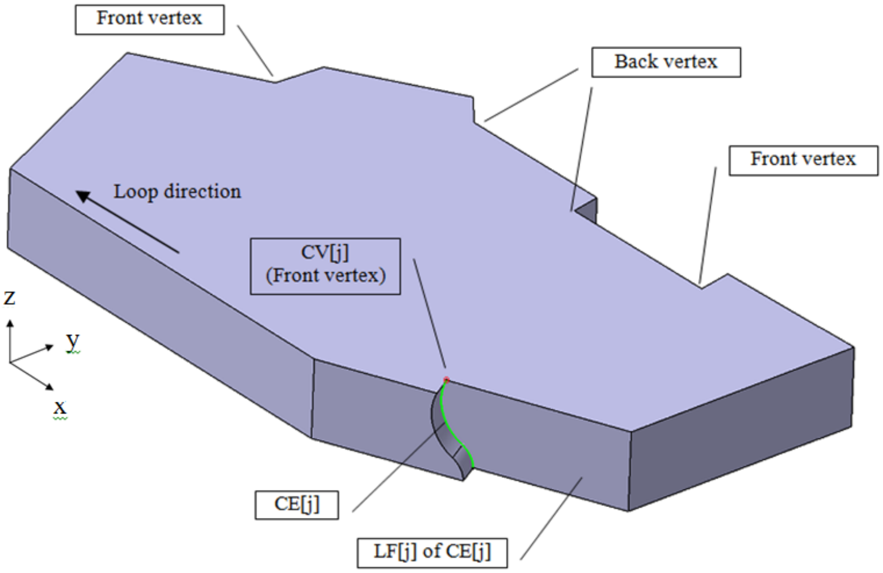


Figure 3-18 Example of front/back concave vertices

Given faces F_1 and F_2 of current DV, their intersections with face F are denoted as E_1 and E_2 respectively. E_1 and E_2 are two of the edges constituting the loop enclosing F, and they do not have common vertex. Denote Q_1 the "to" vertex of E_1 and Q_2 the "from" vertex of E_2. Denote S_1, S_2 and S the surfaces of F_1, F_2 and F respectively. Suppose S_1, S_2 and S intersect at point Q. Surface S_1 and S_2 are able to be trimmed with each other, if the three points determine a triangle ΔQQ_1Q_2 whose area is not zero. That is

$$\left\| \Delta QQ_1Q_2 \right\| \neq 0 \tag{3.21}$$

3.3.8 Island SDV

In this research we define an island SDV as a portion of a delta volume that can be distinguished from other part of the DV by the existence of a self-closed concave edge chain characterizing the intersection of the island's boundary face(s) with other face(s) of the DV. An island SDV represents a "hole" in the part. The "hole" can be through or blind, with regular or irregular

3.3.8.1 Hoop and Island

To avoid confusion with the concept of loop, a self-closed concave edge chain is defined as the "hoop" denoted H. For any face F, if one or some of its edges form a hoop H, then the face(s) sharing H with F is the side face(s) of the island extruding from F. As a hoop H can be completely enclosed by a self-closed curve consisted of interior points of F, it is also an inner loop of F.

An inner loop of a face does not have both concave and convex edges simultaneously. To prove it, suppose two faces CF and XF are boundary faces of an island which is extruded from face F. Denote edge CE and XE the intersections of CF and XF with face F respectively. Thus CE and XE are two of the edges forming the inner loop L of face F. Let $cL = \mathrm{CF} \cap \mathrm{XF}$ so that $\left\| cL \right\| > 0$, otherwise the height of the island is zero, which does not worth discuss. Denote point p the intersection of *cL*, CE and XE. Without loss of generality, suppose CE is a concave edge preceding convex edge XE. In other words, p is the "to" vertex of CE and the "from" vertex of XE. As CE is a concave edge, the island face CF must be on the positive side of F. And face XF bounded by convex edge XE must locate on the negative side of F. So CF and XF can be adjacent only at point p, which is contrary to $\|cL\| > 0$. This contrary indicates the assumption that both concave and convex edges exist in the same inner loop cannot be true. A convex inner loop of a face F does not indicate any volume extruding from F but a “hole” in the face. As a conclusion, an inner loop bounding an island is consisted of concave edges.

Similarly, if an island intersects with more than one faces, it can be proved that all these intersection edges are concave edges, see ANNEX 1.

3.3.8.2 Determination of island SDV

As mentioned above, the side boundary face of an island intersects other face(s) and results in a self-closed concave edge chain, defined as a hoop in this research. An island SDV is determined through the following steps.

Step 1 *Determination of hoop* As each hoop represents the existence of an island, we start at identifying hoops from the edge list of current DV. Algorithm 3.9 is developed for finding out a hoop in a face.

Algorithm 3.9 – Determination of a hoop

```
Given BRep of current DV
For each face f of DV
    L = getloop.inner(f)
       If e[0] of L is concave
              CEList is a hoop H[j]
       End if
End for
```

If an island intersects more than one faces, we have to identify the edges of the hoop one by one. Given a concave edge e1 of face f1, find the concave edge e2 adjacent to e1 at one of its end points, e2 is not an edge of f1. Then find e3, e4 and so forth. If these concave edges form a self-closed chain, then the chain is the hoop to be determined.

Step 2 *Determination of inlaid face* The inlaid face EF of an island is defined as the planar face passing the “lowest” point of the hoop and bounded by the side surfaces of the island. The “lowest” point is the extremum of the hoop with respect to a specific vector **r** determined by Algorithm 3.10, see ANNEX 2. The purposes of determining the inlaid face is to generate a flat end face for the island SDV, so that to create a reference for orientating a tool to remove the SDV, moreover to facilitating the creation of island SDV when it extrudes from several faces that are pairwise adjacent one after another. The inlaid face is also a boundary face enclosing the volume of removable material that will be removed by a single machining operating to produce a correspond “hole” in the part.

Step 3 *Determination of splitting face* The base face(s) BF of the island is defined as the face(s) sharing edges of H with side face(s) of the island. If the base face BF is planar and perpendicular to the vector **r**, the inlaid face EF is then taken as the splitting face spltF. Otherwise a union splitting face has to be created.

Generally when an island extrudes from a curved face or multiple faces, the inlaid face EF determined in step 2 locates inside the current DV, if at least some of the boundary edges of EF are in any existing face, the inlaid face solely cannot fulfill the task to separate an island SDV from current DV, more face(s) needs to be created adjacent to EF, and the splitting face is a union of the inlaid face and these created face(s), see Figure 3-19. Algorithm 3.11 is developed for determining these faces and the splitting face spltF.

Algorithm 3.11

Given base face(s) BF of an island

Given hoop H of the island

Given the ECList of the island

Given the surface list SFList of the island

Given inlaid face EF of the island

Make new side face(s) NSF by trimming each member of Slist with the members adjacent to it, and with EF and ECList

```
        Make the splitting face spltF by union BF with NSF
//The island SDV is bounded by NSF, EF and face(s) on the right side of ECList
End of Algorithm 3.11.
```

Step 4 *Determination of island SDV and renewing DV* The island SDV[i+1] is obtained by splitting current DV[i] with the splitting face spltF[i+1] determined in step 3. Two different cases have to be take into account while updating DV[i] to DV[i+1].

First, if the base face BF is planar and perpendicular to the vector **r**, the inlaid face EF is co-surfaced with BF in the same plane, and EF is the only splitting face. Taking the island SDV away will leave a planar boundary face to the new DV. In this case, DV[i] is updated to DV[i+1] simply by using equation (3.3).

Second, at least one of the conditions given above does not satisfied, the removal of SDV[i+1] will leave a pocket or hole in DV[i+1], as if a portion of the island SDV[i+1] is embed in DV[i], see Figure 3-19.

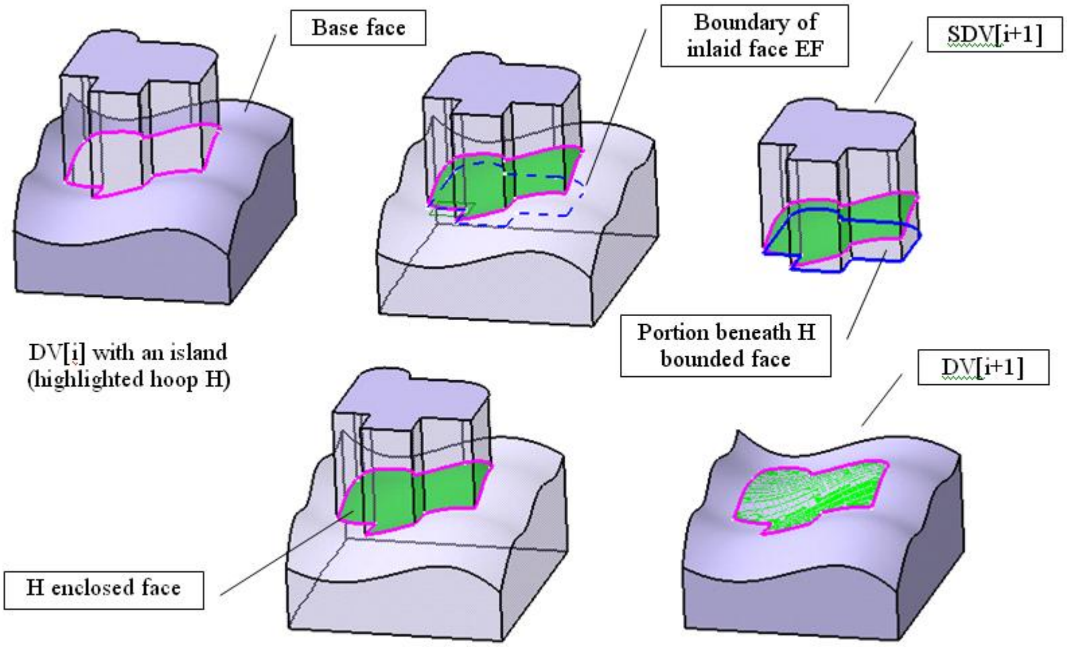

Base face
Boundary of inlaid face EF
SDV[i+1]
DV[i] with an island (highlighted hoop H)
Portion beneath H bounded face
DV[i+1]
H enclosed face

Figure 3-19 An example of island SDV and DV update

To avoid leaving a complicated shape to updated DV[i+1], we develop an algorithm for filling the pocket/hole left to the remaining volume of material as an island SDV is subtracted. To achieve this purpose, instead of subtracting the full volume of SDV[i+1], we take off only the portion of defined SDV[i+1] that extrudes out of the base face(s). Thus created face(s) bounded by the hoop performs the splitting for updating DV[i] to DV[i+1]. Although this treatment causes duplicating the volume of material, which is equal to the portion of the island SDV sinking in current DV, it will simplify the tool path. Figure 3-19 shows an example of creating an island SDV[i+1] extruding from a curve face, and the update of current DV[i] to DV[i+1].

Semi-island, even possessing a face overlapped with the block face, as illustrated in Figure 3-20, can be separated from a current DV with the algorithms to be introduced in following sub section. Hence no detail discussion will be made especially on this issue.

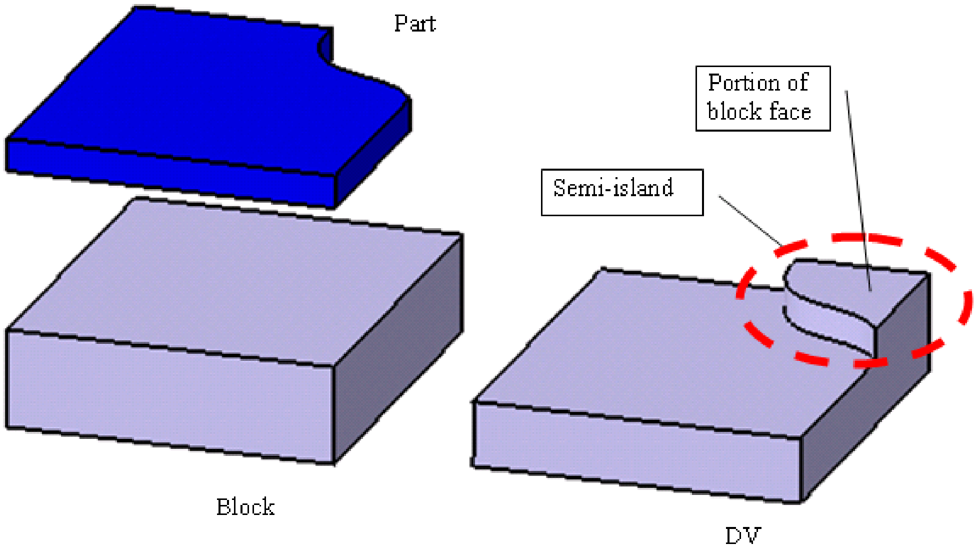


Figure 3-20 An example of semi-island in a DV

3.3.9 Interior DV decomposition

By determine the facing SDV and exterior SDVs, material surrounding the part P are divided into smaller volumes each is intended to be removed by one milling operation, and the remaining removable material are mainly constrained within the space enclosed by faces of the part and previous created splitting faces. Thus the remaining volume of DV[0] is named interior DV.

An interior DV needs to be decomposed at locations where two or more faces intersect forming a concave edge, because each of these concaved edges indicates a sharp change of the part’s shape that may lead to changes in tool parameters, tool trajectory, or machining strategies. Thus a SDV ought to be separated from a current DV at each of these locations. The decomposition begins at detecting concave edges in the enclosure faces of current interior DV.

In this work, the concave edge of a “hole” in the DV is not considered as a location for decomposing, because it signifies an extrusion in the part, and for simplification, we suppose material around such an extrusion is removed by a tool traveling around it.

3.3.9.1 Steps of decomposition

Different from island SDV, one of the common characteristics of interior SDVs is that their concave edge chains, instead of forming a hoop, are not self-closed. The ends of each of these chains can terminate at either the same face or different faces. In both cases the vertex corresponding to an end of the concave edge chain must be a concave vertex in the loop enclosing the face where the chain comes to an end. The face terminating a concave edge chain does not consist of any edge of the chain. The Approaches for determining this kind of extrude SDV are consisted of the following steps.

Step 1 *Find out concave vertex* For each created face F[m] of current DV, get the outer loop L[m] of F[m]. Starting arbitrarily at a vertex of the loop, for example V[0] of e[0] of L[m], and following the direction of the loop, verify the concavity of each vertex. Put all concave in a list denoted CVlist[m].

Step 2 *Form concave edge chain* For each concave vertex CV[j] in CVlist[m] find out the concave edge CE[j] passing CV[j] but not belonging to L[m]. Obviously CE[j] correspond to one of the part edges, as the previous decompositions do not create any new concave edge in a DV. For each CE[j], detect all concave edges $\{e_i\}$ that are adjacent one with the other, make a concave edge chain Echain[j] by join edges of $\{e_i\}$ together.

$$\mathrm{Echain}[\mathrm{j}] = \bigcup \{e_i\} \tag{3.22}$$

If one of the concave edges ends at point Q which is a vertex of an inner loop inL of the face enclosed by L[m], one or more edges of the inner loop can be added to the chain. Denote $\{E_k\}$ the set of edges of the inner loop to be added. Edges of $\{E_k\}$ are continuously adjacent one to the other, see Figure 3-21. Remember the direction of an inner loop is counter clockwise, hence the “to” vertex of edge $E_1 \in \{E_k\}$ is the point Q. E_2 is preceding to E_1 in the inner loop inL, E_3 preceding to E_2 and so on. The last edge $E_K \in \{E_k\}$ starts at vertex V_K which is the first concave vertex preceding Q in inL. There is a concave edge cE passing V_K. Add cE and other adjacent concave edges, if there is any, to the chain one after another until no adjacent concave edge can be found, and the chain ends at a concave vertex.

The vertex CV[j] used to determine Echain[j] = $\{e_i\}$ is the start point of the chain. In other words it is the “from” vertex of the first edge $e_1 \in \{e_i\}$ of the chain.

Figure 3-22 shows an example of the outer loop of the top face of current DV which is an interior volume (all surrounding exterior SDVs have been separated). The concave vertices and concave edge chains are highlighted in this figure.

In the proposed approach, decomposition terminates when no concave point can be found in the outer loop of every current DV face.

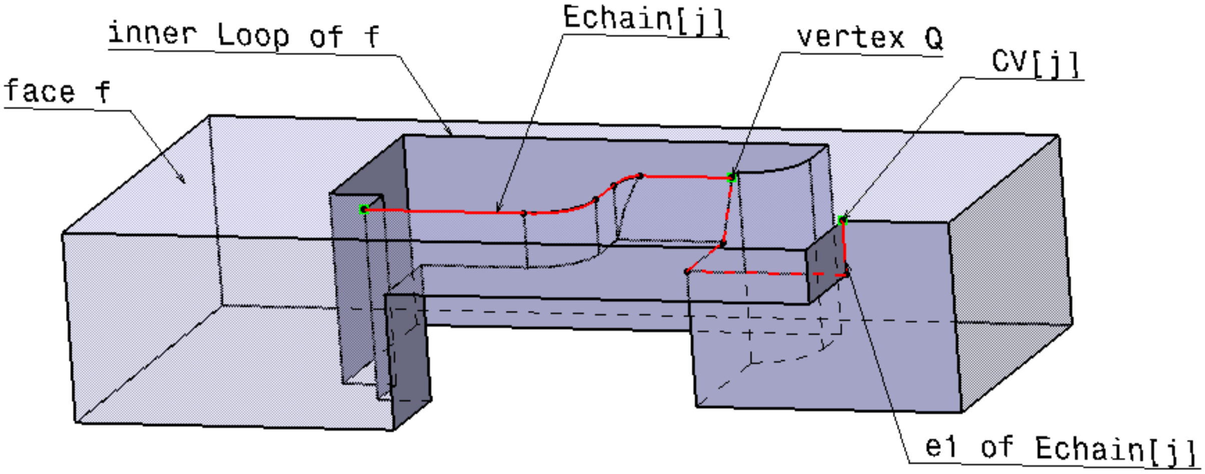


Figure 3-21 Example of Echain[j] meets vertex Q of the inner loop of *f*

Step 3 *Generate splitting face* A splitting face is generated for each Echain[j]. Corresponding to direction **n**, the first splitting face will be created for the chain, not lose generality denote it Echain[j], whose start vertex is the farthest in the direction **n** as defined by

$$\mathrm{CV[j]} = \max\left(\{\mathrm{Echain[i].vertex}\},\ \mathbf{n}\right) \tag{3.23}$$

where $\{\text{Echain[i].vertex}\}$ is the set of vertex of all concave chains.

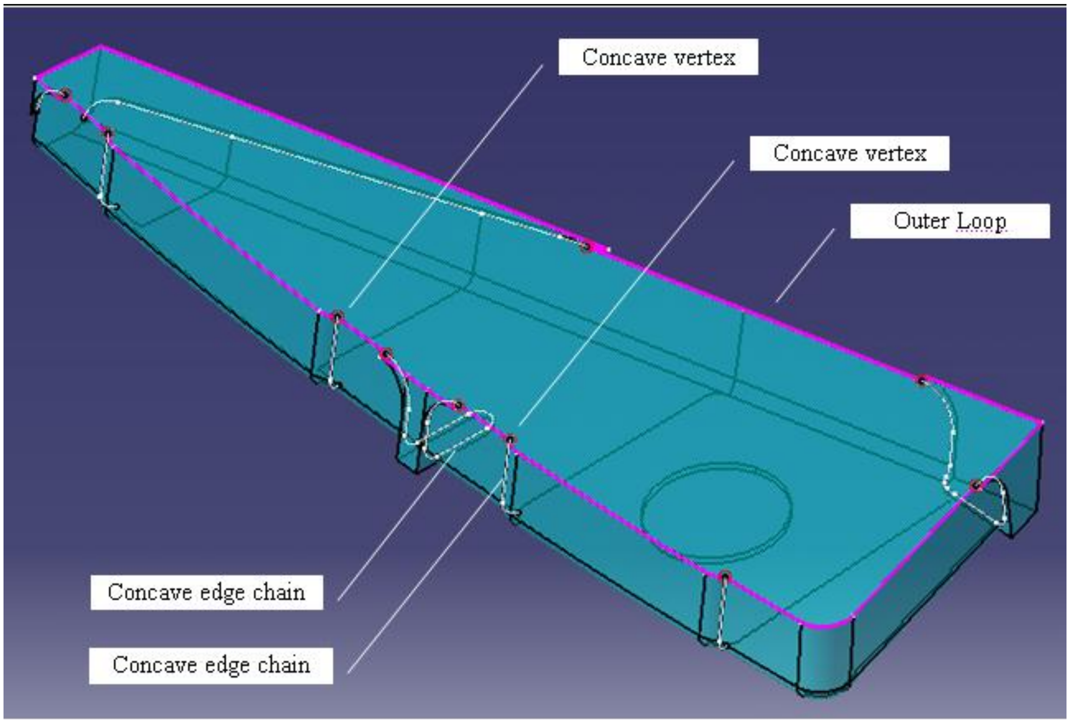


Figure 3-22 Concave vertices and concave edge chains

3.3.9.2 Creation of splitting faces

Four cases are considered in creating a splitting face for separating an interior SDV. To save space we assume that, except edges of Echain[j], no other concave edge intersects surface LS[j]. Otherwise we can find out these intersected concave edges and their left and right surfaces, and let the split face to be created be bounded by those surfaces.

Case 1 All edges of the chain Echain[j] $=\{e_i\}$ are co-surfaced, as illustrated in Figure 3-23. For the concave vertex CV[j] corresponding to Echain[j] find out the closest side face of the stock sF. Denote **n** the normal of sF, suppose $e_1 \in \{e_i\}$ has vertex CV[j] as its "from" vertex, denote LF and RF the left and right faces of e_1 respectively. Denote LS and RS the surfaces of LF and RF respectively. Denote **n**LF and **n**RF the normal vectors of LF and RF at CV[j] respectively. The surface S[j] to be used for generating a splitting face corresponding to the edge chain is determined as follows

If $\{e_i\} \in \mathrm{LS} \wedge \{e_i\} \notin \mathrm{RS}$, and $\mathbf{n}\mathrm{LF} \cdot \mathbf{n} > \mathbf{n}\mathrm{RF} \cdot \mathbf{n}$,

$$\mathrm{S}[\mathrm{j}] = \mathrm{LS} \tag{3.24}$$

vice versa;

If $\{e_i\} \in \mathrm{RS} \wedge \{e_i\} \notin \mathrm{LS}$, $\mathbf{n}\mathrm{LF} \cdot \mathbf{n} > \mathbf{n}\mathrm{RF} \cdot \mathbf{n}$,

$$\mathrm{S}[\mathrm{j}] = \mathrm{RS} \tag{3.25}$$

The condition $\mathbf{n}\mathrm{LF} \cdot \mathbf{n} > \mathbf{n}\mathrm{RF} \cdot \mathbf{n}$ in equation (3.24) is for creating a splitting surface closer to the wall faces corresponding to direction **n**.

One situation is considered that CV[j] and CV[k] are front and back vertex respectively, Echain[j] $=\{e_i\}$ is co-surfaced chain, and each edge of $\{e_i\}$ has its corresponding edge in Echain[k] $=\{ec_i\}$. The splitting face for this situation will be discussed in case 3 and 4.

A splitting face can be generated according to the following situations: 1) CV is the sole concave vertex, 2) all other concave vertices are located on the negative side of LS or RS, 3) some concave vertices are located in the positive side of LS or RS. Not lose generality suppose they are located in the positive side of LS.

In situation 1) and 2) the splitting face is determined by

$$\mathrm{spltF} = \mathrm{intersect}\left(\mathrm{DV}, \mathrm{LS}\right) - \mathrm{LF} \tag{3.26}$$

While in situation 3) a splitting face should be created in such a way that it will separate a SDV without any concave vertex from current DV. This objective can be achieved by the following steps.

Step one, detect the concave vertex CV[k] which is first vertex succeeding CV[j], as they are sequenced in the set $\{\mathrm{CV[i]}\}$; not lost generality suppose CV[k] is a back vertex.

Step two, denote $\mathrm{Echain[k]} = \{ec_i\}$ the chain of concave edges starting at CV[k], detect the point *p* by

$$p = \max\left(\left\{ec_i\right\}, \mathbf{n}\mathrm{RF}\right) \tag{3.27}$$

p is the point, on an edge of Echain[k], that is the nearest to the right surface RS of the first edge of Echain[j]. Denote $ec_k \in \left\{ec_i\right\}$ the edge where point p is detected, denote LS[k] and RS[k] the left and right surfaces of ec_k respectively.

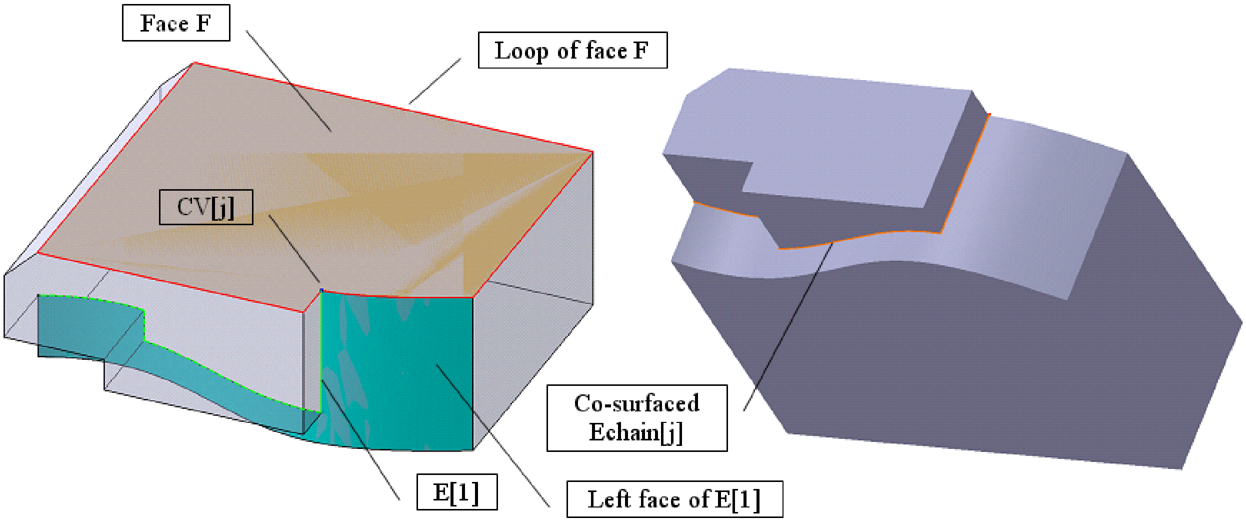


Figure 3-23 An example of Case 1

Step three, if within the domain of the stock $\mathrm{LS} \cap \mathrm{LS[k]} = \varnothing$, or even when $\mathrm{LS} \cap \mathrm{LS[k]} \neq \varnothing$, but $\mathrm{LS[j]} \cap \mathrm{LS[k]} \cap \mathrm{DV} = \varnothing$, which means that the two left surfaces intersect outside the DV or do not intersect at all, then determine the splitting face as follows

$$\mathrm{spltF[j]} = \mathrm{intersect}\left(\mathrm{DV}, \mathrm{LS}\right) - \mathrm{LF} \tag{3.28}$$

If $\mathrm{LS} \cap \mathrm{LS[k]} \cap \mathrm{DV} \neq \varnothing$, which means that the two surfaces intersect inside the DV, the splitting face spltF[j] is defined as the portion of the intersection of LS with current DV limited by LS[k] and Echain[j]. Denote $\mathrm{spcN}\left(\mathrm{LS}\left[\mathrm{k}\right]\right)$ the negative half space of surface LS[k], then we have

$$\mathrm{spltF[j]} = \mathrm{intersect}\left(\mathrm{DV}, \mathrm{LS}\right) \cap \mathrm{spcN}\left(\mathrm{LS}\left[\mathrm{k}\right]\right) - \mathrm{LF} \tag{3.29}$$

As an example, the volume given in Figure 3-23 is decomposed, and the result is illustrated in Figure 3-24.

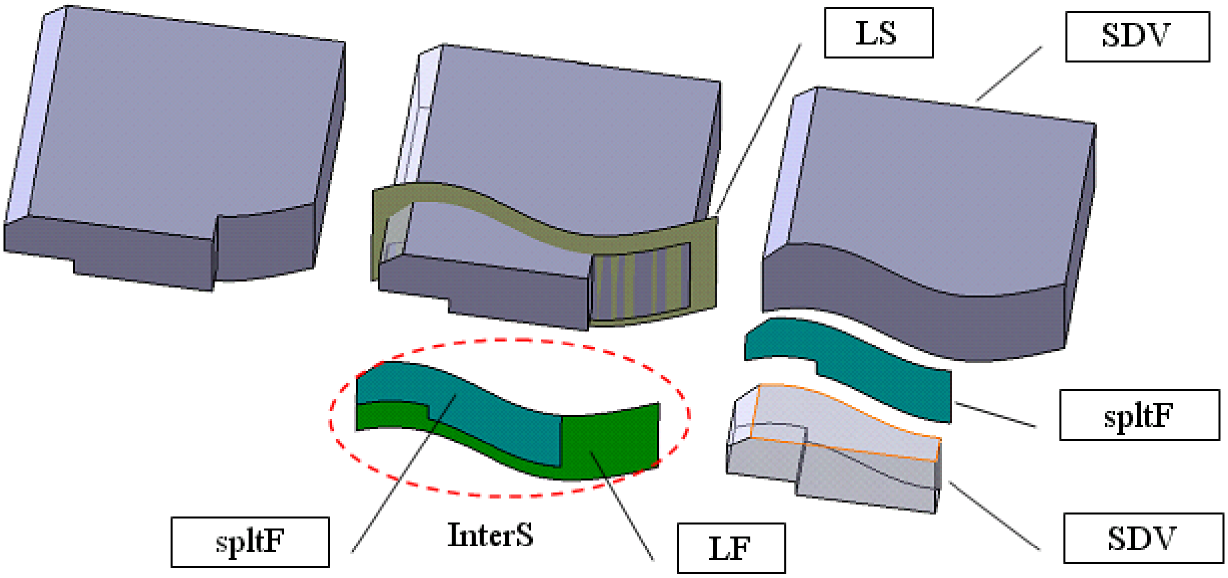


Figure 3-24 An example of Case 1

Algorithm 3.12, see ANNEX 3, is developed to determine the splitting face of a co-surfaced edge chain.

Case 2 Not all edges of Echain[j] are co-surfaced, as illustrated in Figure 3-25, where LS[1] is the left surface of the first edge E[1] of Echain[j], but both end vertices of Echain[j] are in the loop L, and LS[1] is able to be trimmed by RS[N], where RS[N] is the right surface of the last edge E[N], further more at least each pair of edges $\{\mathrm{E}[\mathrm{i}], \mathrm{E}[\mathrm{N}-\mathrm{i}+1]\}$, $\left(1 \le \mathrm{i} \le \left\lfloor \frac{N}{2} \right\rfloor\right)$ of the chain are co-surfaced in LS[i]. Remember that when a chain is referenced, the orientation of an edge in the chain is defined corresponding to the direction of the chain, which is from the first edge to the last edge of the chain. Under these conditions, without loss of generality, suppose CV[j] is a front vertex in L, a face piece pF[i] in LS[i] is bounded by each pair of co-surfaced edges, E[i] and E[N-i+1], and the intersections of LS[i] with LS[i-1] and LS[i+1]. If there are an odd number of edges in a chain, and if three edges in the middle of the chain are co-surfaced, then these three edges together are used to determine a face piece. Otherwise, the face piece corresponding to the middle edge is the intersection of its left surfaces with that of those adjacent to it.

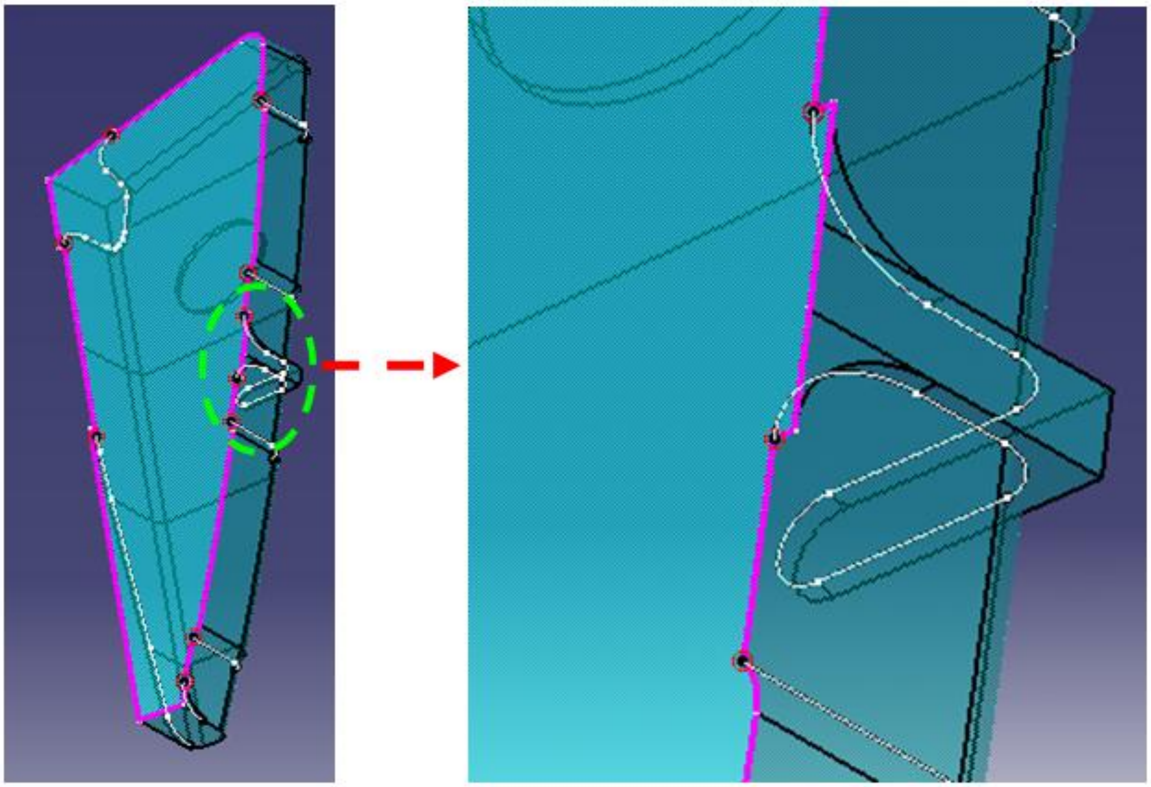


Figure 3-25 An example of Case 2

If the two edges of a pair, for example edge pair E[i] and E[N-i+1], are not co-surfaced in the specified left surface, but the left surface LS[i] of E[i] is able to be trimmed by the right surface RS[N-i+1] of the coupled edge E[N-i+1], then the portion of LS[i] bounded by four surfaces: LS[i-1], LS[i+1], RS[i] and RS[N-i+1] is taken as the face piece as referenced by the edge pair E[i] and E[N-i+1].

The situation that LS[i] cannot be trimmed by RS[N-i+1] will be discussed in Case 4. Algorithm 3.13, see ANNEX 4, is developed for determining the face pieces of Case 2. When all face pieces are defined, a union of them can be made to create the splitting face for the corresponding concave edge chain Echain[j].

We can also group the edges of Echain[j] in such a way that adjacent edges possessing the same left face will form a chain segment. Thus instead of pairing edges, chain segments are coupled.

<u>Case 3</u> CV[j] is the only vertex of Echain[j] located in the outer loop L. Without loss of generality, assume that CV[j] is a front concave vertex and CV[j+1] is a back concave vertex as defined in section 3.3.7, they belong to different concave edge chains $\mathrm{Echain}[\mathrm{j}]=\{e_i\}$ and

$\text{Echain}\left[\text{j+1}\right]=\left\{ec_i\right\}$, see Figure 3-26. In this situation, there is extrude removable volume between Echain[j] and Echain[j+1]. It is further assumed the left surface LS[j] of edge $e_1 \in \left\{e_i\right\}$, is able to be trimmed by the left surface LS[j+1] of $ec_1 \in \left\{ec_i\right\}$. In most cases, edges of Echain[j] and Echain[j+1] can be coupled in the same order as the edges are ordered in the chains, see Figure 3-26. Face pieces corresponding to individual pairs of edges can be created in a way similar to what we have discussed above in case 2, details will not be repeated.

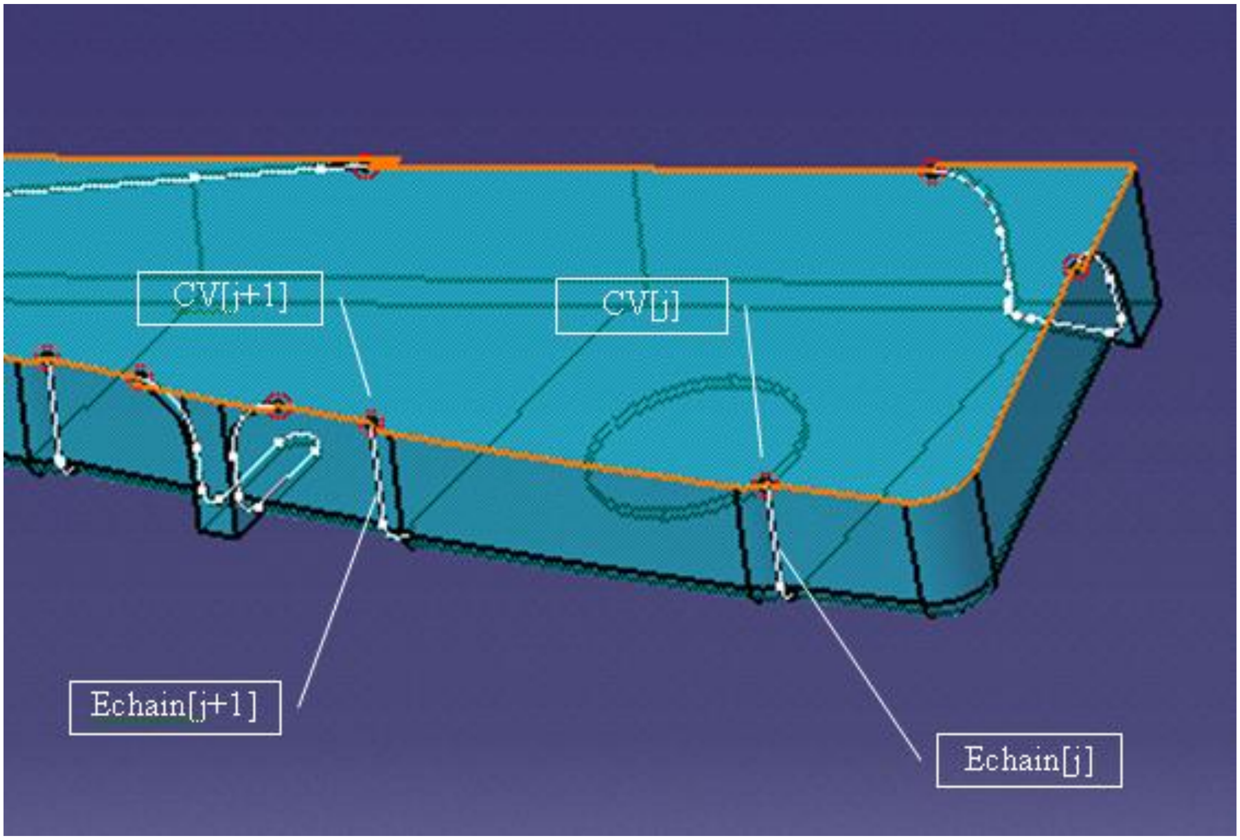


Figure 3-26 An example of Case 3

In the exception that there are unequal numbers of edges in Echain[j] and Echain[j+1], but both of these chains can be divided into the same number of chain segments, each segment is consisted of co-surfaced edges. In this case, instead of matching edges, we can couple the chain segments in such a way that one segment is chosen from each of these chains, and then join the ends of coupled chain segments with line segments. Individual face pieces are bounded respectively by loops each consisted of a coupled chain segments and the corresponded line segments.

In case the numbers of chain segments are not equal, chose the chain consisted of more edges, the chain $\{e_i\}$ for example. Without loss of generality suppose that $\{e_i\}$ starts at a front vertex. Face piece f_j corresponding to edge $e_j \in \{e_i\}$ is determined by

$$f_j = \text{trim}\left(\text{LS}_j,\ \text{LS}_{j-1},\ \mathbf{r}_{j-1}\right) \cap \text{trim}\left(\text{LS}_j,\ \text{LS}_{j+1},\ \mathbf{r}_{j+1}\right) \cap \text{spcN}\left(\text{RS}_j\right) \cap \text{spcN}\left(\text{Ls}_j\right) \quad (3.30)$$

where Rs_j is the right surface of the edge $ec_j \in \{ec_i\}$ corresponding to $e_j \in \{e_i\}$, $\text{spcN}\left(\text{RS}_j\right)$ and $\text{spcN}\left(\text{Ls}_j\right)$ are negative half spaces of RS_j and Ls_j respectively, trim(objA, objB, **r**) is the function performing trimming objA with objB, and as a result return the portion of objA in the half space of objB indicated by **r**. In equation (3.30) vector $\mathbf{r}_{j-1}$ and $\mathbf{r}_{j+1}$ are determined by

$$\mathbf{r}_k = \left(j-k\right)\left(\mathbf{n}\text{LF}_k \cdot \mathbf{n}\text{LF}_j\right)\mathbf{n}\text{LF}_k \qquad (3.31)$$

which is the indicator of the half space in which the portion of the trimmed object should be kept. Vector $\mathbf{n}\text{LF}_k$ is the normal of the left face LF_k of $e_k \in \{e_i\}$ at point $p_k = e_j \cap \text{LF}_k$.

Denote N the number of edges of $\{ec_i\}$ $\left(i = 1,\dots,\text{N}\right)$, when $j \geq \text{N}+1$, all face pieces f_j will be trimmed by Ls_N as there is not corresponding Ls_j. The first and the last face piece will be trimmed respectively by the surfaces corresponding to faces sharing the start and end vertices of $\text{Echain[j]} = \{e_i\}$.

<u>Case 4</u> Under the circumstance that LS[j] is not able to be trimmed by LS[j+1], as illustrated in Figure 3-27, no matter CV[j] and CV[j+1] are in the same concave edge chain or not, we can

divide Echain[j] into segments, each segment is consisted of adjacent edges bounding the same left face. For each segment Sechain[k] of Echain[j], a face piece is defined as the portion of the intersection of current DV and the left surface LSS[k] of any edge of Sechain[k]. The portion is limited by Sechain[k] and two intersections: LSS[k] and LSS[k-1], LSS[k] and LSS[k+1]. For the first segment Sechain[1] of Echain[j], the corresponding face piece is the portion of the intersection of surface LSS[1] with DV limited by the intersection of LSS[2] and Sechain[1]. Analogously, for the last segment chain Sechain[K], where K is the number of segments that

Echain[j] is divided, the splitting face is the portion of the intersection of DV and LSS[K] bounded by LSS[K-1], and Sechain[K].

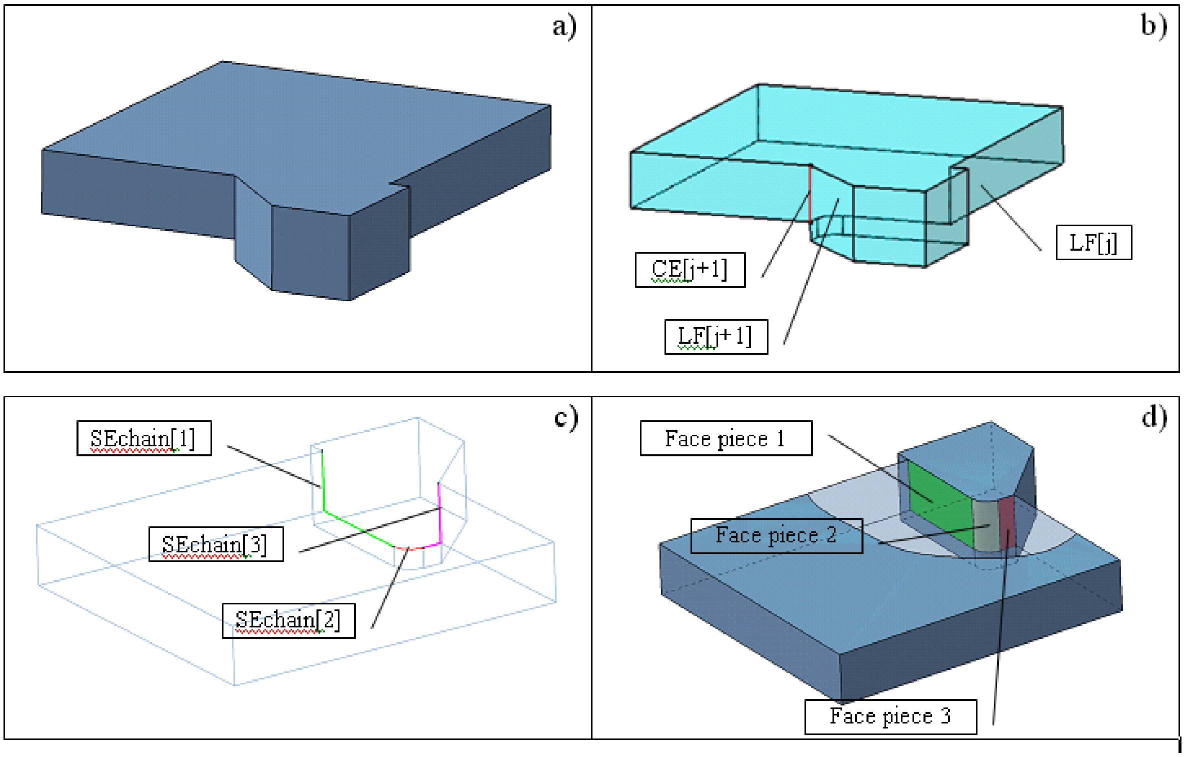


Figure 3-27 An example of Case 4

Case 2 through 4 give the Approaches for generating face pieces. Once all face pieces in accordance to an Echain are created, a union of these faces UF can be generated for splitting a SDV from current DV.

Generally, the positive direction of the splitting face is corresponding to the normal vectors of the face pieces. But the direction is reversed, if the chain Echain ends at an edge belonging to a former inner loop. The loop does not exist at the moment when UF corresponding to Echain is to be applied to separate a SDV, because a previous splitting operation consisting one or more edges of the loop has broken it, as shown in Figure 3-28.

Step 4 *Determine SDV* Use the splitting face spltF = UF to cut current DV[i] and get a new sub delta volume SDV[i+1], update DV[i] to DV[i+1] as defined in equation (3.3). In the situation of case 4 the splitting is conducted one by one in accordance with the order of the face groups.

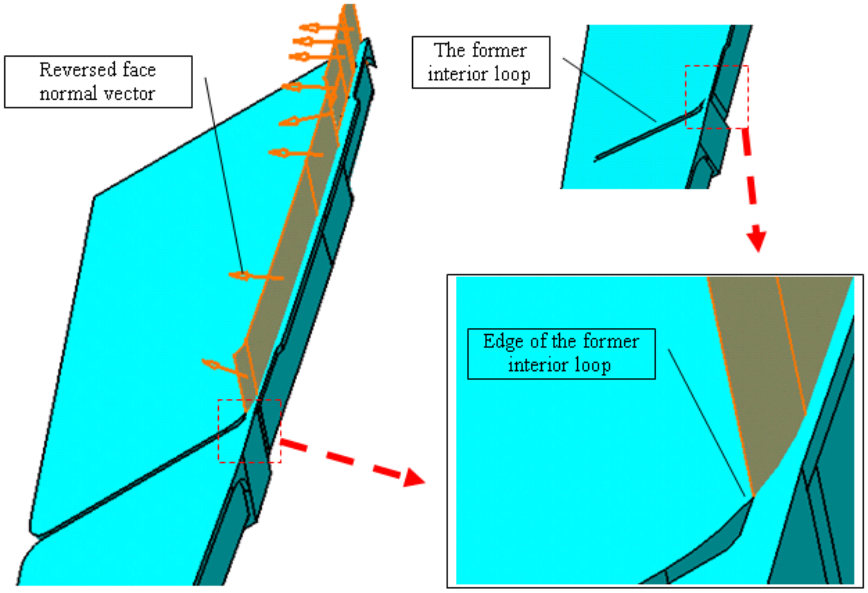


Figure 3-28 An example of reversing positive side of a UF

Figure 3-29 illustrates the results of decomposing interior volume of a sample part, where SDV[13] is an island SDV. The other SDVs are split from corresponding DVs with splitting faces determined by concave edges.

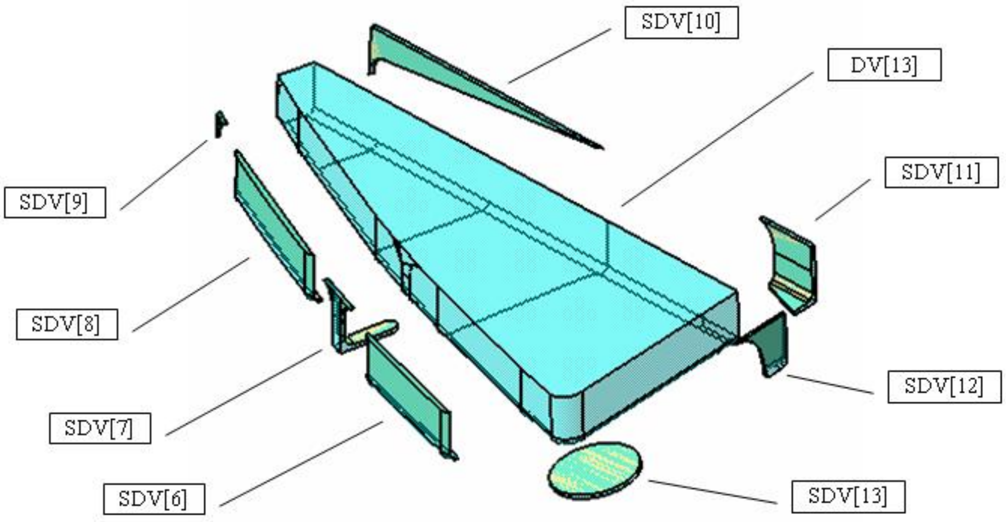


Figure 3-29 Decomposition of interior volume

3.4 An example of multi-pocket

In this section an aerospace structural part, Part II, with multi-pocket is taken as an example to evaluate our algorithms of delta volume decomposition. As shown in Figure 3-30, this part has two semi closed pockets bounded by inclined curved wall faces. On the bottom of these pockets, there are several through holes which are corresponding to island SDVs from the delta volume decomposition perspective. The largest face of the part is planar, which is chosen as the support face and is overlapped with one of the stock faces. The relative positions of the part P and the stock are demonstrated in Figure 3-31.

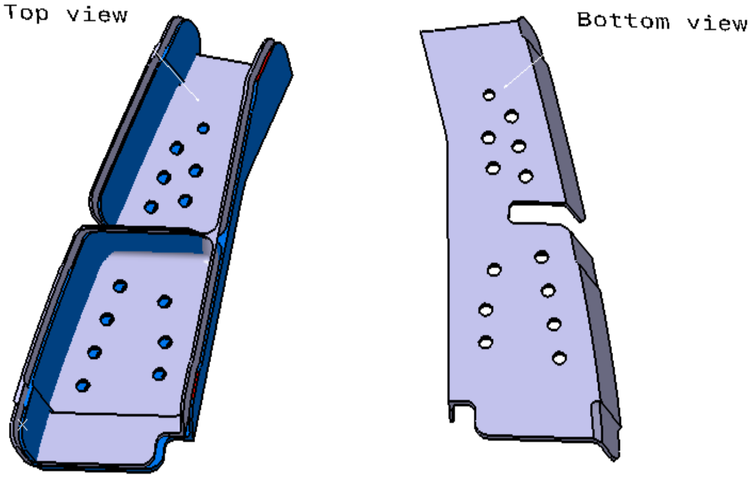


Figure 3-30 Part II - an aerospace structure part

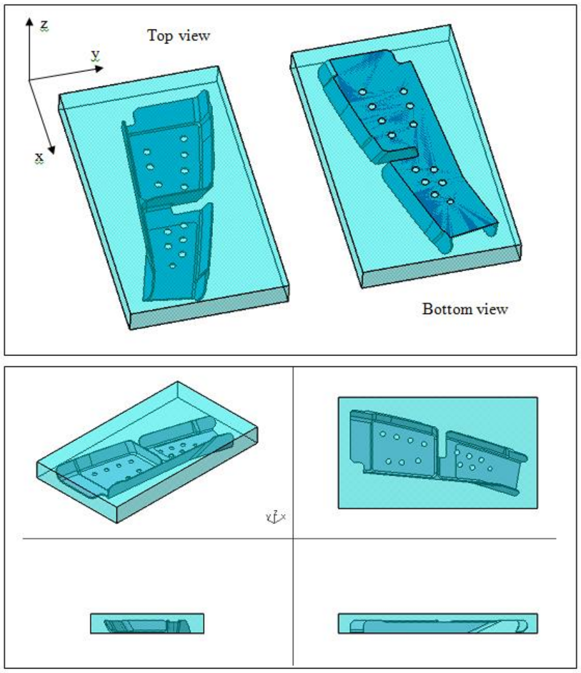


Figure 3-31 Relative positions of part II and the stock

The initial delta volume DV[0] obtained by regularized Boolean subtraction of the part P from the stock S is illustrated in Figure 3-32. In the upper part of this figure, sections of DV[0] are given. In the lower part of this figure, DV[0] is demonstrated in multi-view style.

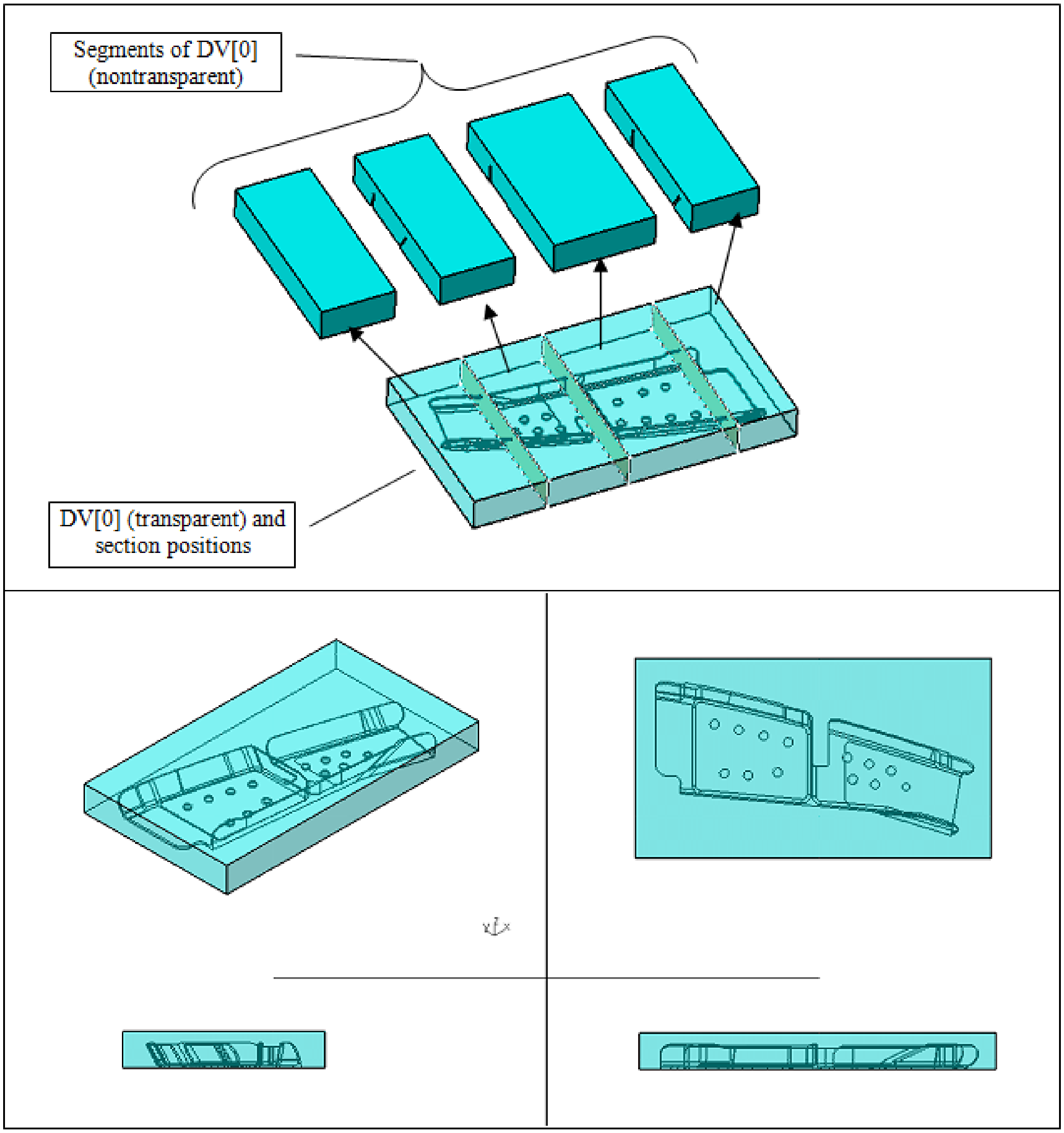

Segments of DV[0] (nontransparent)
DV[0] (transparent) and section positions

Figure 3-32 The initial delta volume DV[0] of sample part II

The decomposition procedures commence at detecting the extremum of the part in positive z direction. In this example, two planar face extrema are found, as shown in Figure 3-33. A planar splitting face spltF[1] passing these extrema is created for separating the facing sub delta volume

SDV[1]. In Figure 3-34 a) shows DV[0], b) through d) illustrates the processes of creating spltF[1], separating SDV[1] from DV[0], and as a result a new delta volume DV[1] is obtained. Figure 3-34 e) through h) show the comparison between DV[0] and DV[1] in multi-views.

Individual exterior sub delta volumes, SDV[2] to SDV[5], shown in Figure 3-35, are separated from related delta volumes, DV[1] to DV[4], by splitting faces which are generated according to the properties of the extrema of the part in specific directions.

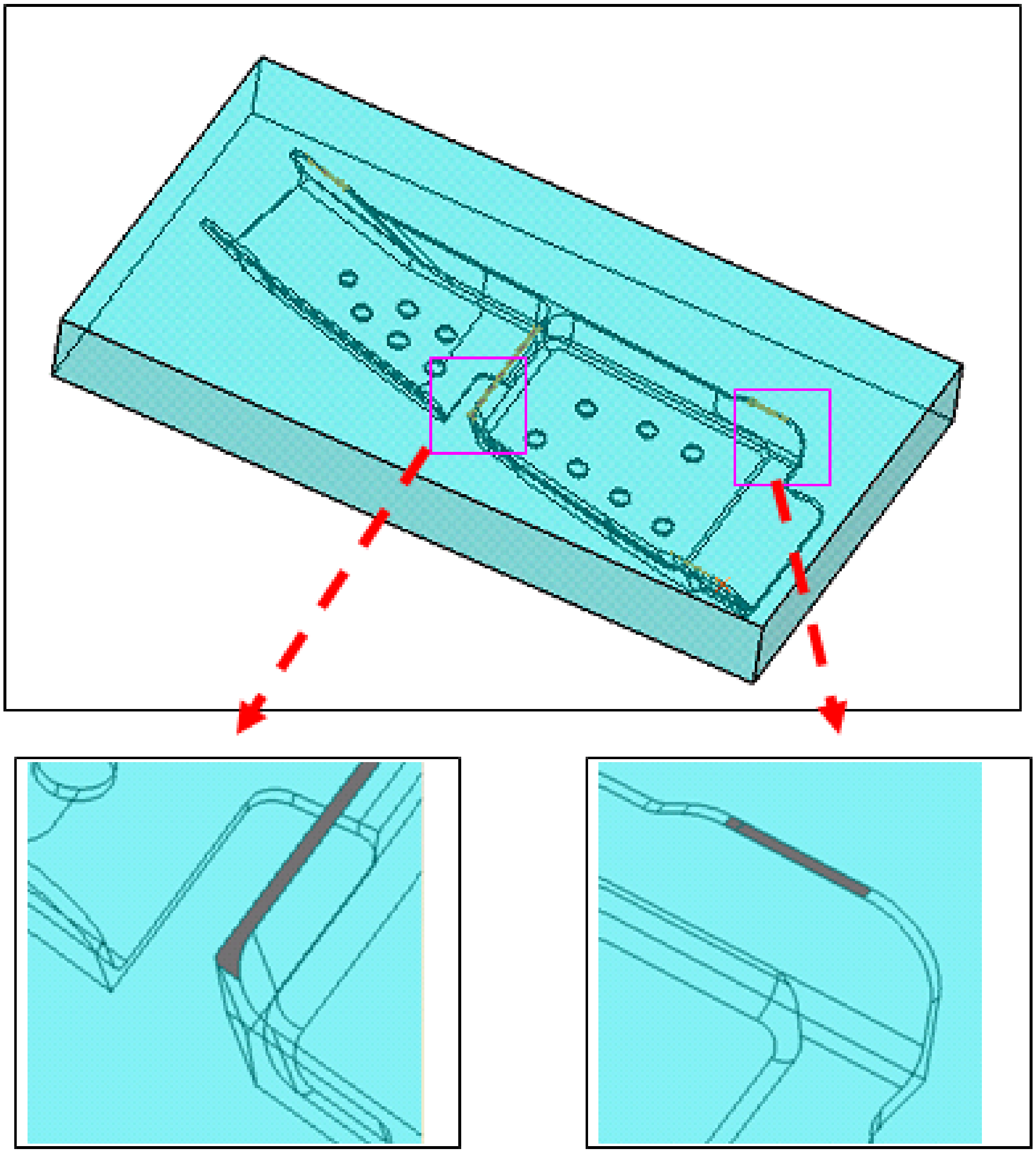


Figure 3-33 Extremum in positive z direction of sample part II

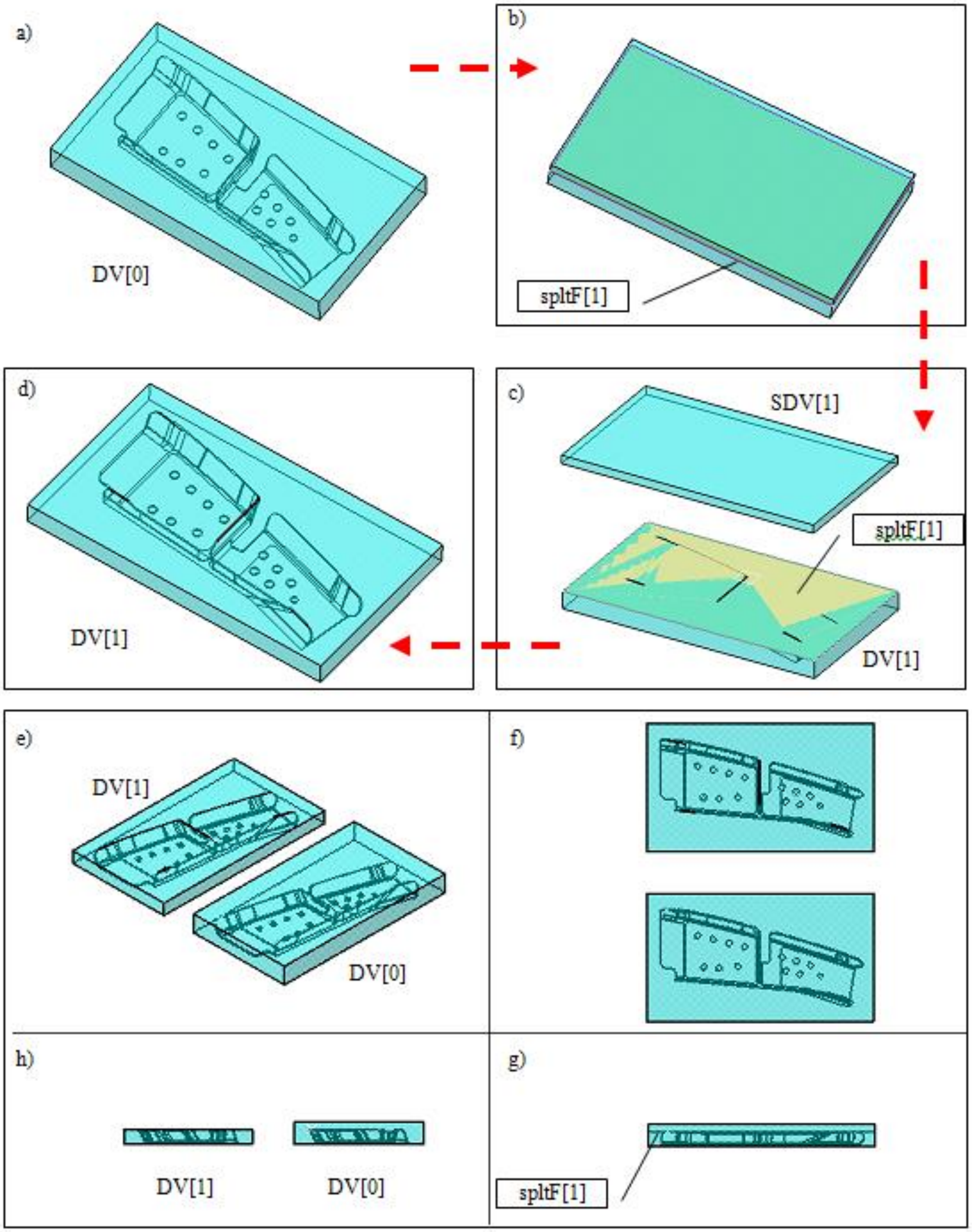


Figure 3-34 Separating SDV[1] from DV[0]

Island SDVs, SDV[6] to SDV[18], are also illustrated in Figure 3-35.

The decomposition procedures are indicated step by step with red arrows.

Figure 3-34 also illustrates step by step the changes of DVs as exterior SDVs (SDV[2] through SDV[18]) are separated one after another.

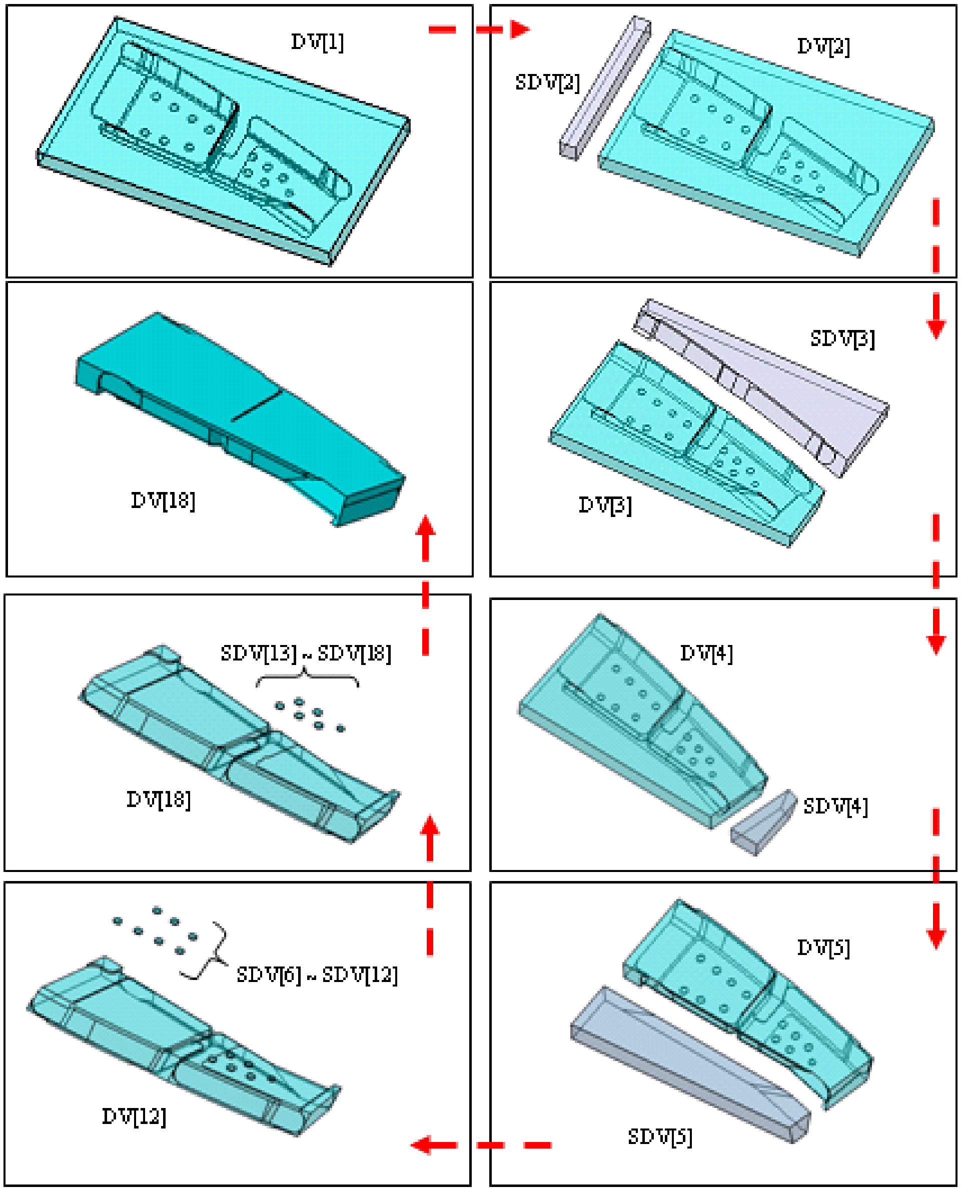


Figure 3-35 Decomposition of exterior and island SDVs of sample part II

The thirteen island SDVs (SDV[6] through SDV[18] are exterior SDVs because their end faces are subsets of the bottom face of the stock.

In the outer loop of the top face of DV[18], which is the primary status of the interior delta volume, there are several concave vertices, as illustrated in Figure 3-36. Some of them are front concave vertices, P2 and P4 for example, and others are back vertices, such as P1 and P3. In this example, the interior volume decomposition commences at the concave vertex P1 which has the smallest y-coordinate as compared with other concave vertices in the outer loop.

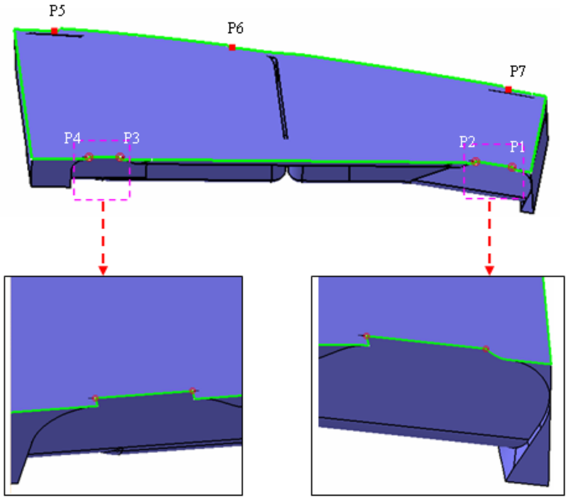


Figure 3-36 Concave vertices in the outer loop of the top face of DV[18]

The result of complete decomposition of the interior delta volumes is illustrated in Figure 3-37. The last sub delta volume SDV[30] is exactly the same as delta volume DV[30] which does not have any concave vertex in any boundary face.

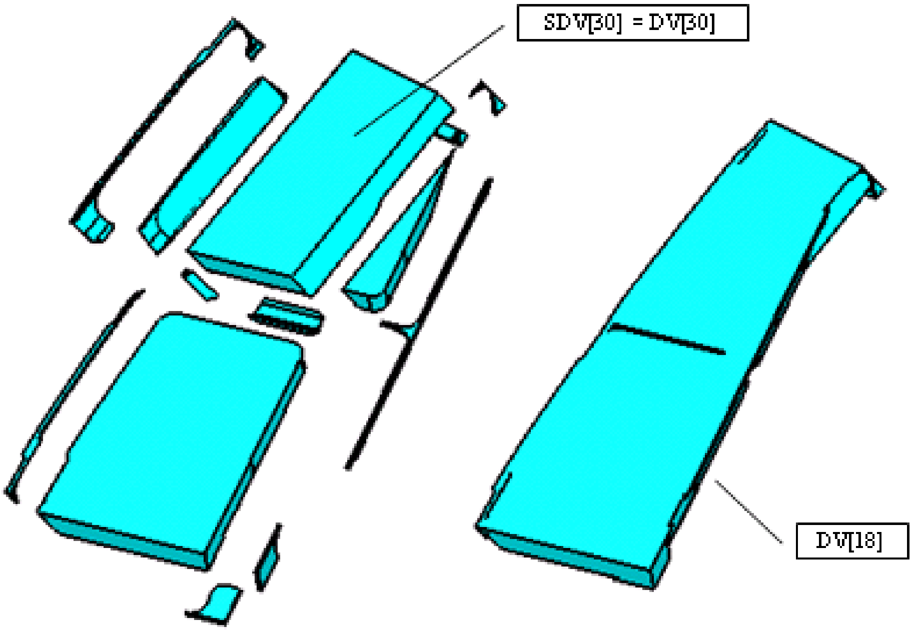


Figure 3-37 Completed decomposition of sample part II

CHAPTER 4 DETERMINATION OF MILLING ATTRIBUTES

Chapter 3 provides algorithms that can be used to decompose delta volumes into sub delta volumes SDVs. The purpose of delta volume decomposition is to determine the geometries of individual removable volumes of material. Based on the results of delta volume decomposition, in this chapter, we will define milling attributes of an SDV, to develop approaches for identifying and extracting these attributes so that information that is meaningful to machining can be associated with individual SDV models.

4.1 Machining attributes

An SDV represents a specific volume of material to be removed by a machining operation. To remove an SDV, some essential decisions about where a tool can start cutting the material, how should the tool be oriented initially, what type and dimensions of a tool should be selected etc., have to be made in programming process. In this research the following information about a SDV are referred to as the machining attributes of the SDV.

1) Access face, a face which is accessible to a tool from the exterior of the SDV
2) Access point, a position where a machining operation can start
3) Tool orientation, directions along which a tool can be oriented
4) Tool geometry (diameter, length, tip shape, etc.)

Here the “access position” is a point on the boundary of a SDV where a tool can start breaking the boundary and removing the material of specific SDV. Although radial and axial compensation have to be considered in determining the initial position of a tool, at this stage they are ignored because a tool is simplified as a ray (semi-line) with one sole end point coincident with the center point of its tip. Radial and axial compensations can be added to the initial position according to the specific parameters of a selected tool later.

Tool axis initial orientation is represented by a vector indicating the direction to which the tool axis is parallel during roughing operation applied on a specific SDV. In many cases a proper initial direction allows a tool finish roughing a SDV without changing its axis direction.

Geometries of a SDV and its special position relative to other SDVs have critical effects on the determination of tool parameters such as diameter, cutting length, tip shape, etc.. Usually a SDV may require different tool parameters when accessed from different directions.

It is evident that all these attributes are determined not only by geometries of the SDV in question, more factors have to be taken into account, such as characteristics of faces/edges of a SDV, the position of the SDV relative to other SDVs that will be removed in a operation succeeding. Comprehensive analysis is required to determine the machinability. A part is machinable with a specific machine tool only when all of its machining attributes do not exceed corresponding processing capacities of a given machine.

4.2 Determination of access position

As defined above, tool access position is a point on the boundary of a SDV at which the tip center of a tool can be located for commencing a machining operation. For description simplification, the tool radius compensation is not taken into account at this moment. As a tool will cut into the volume of material from this point, thus the access position must situate either in the interior of a removable face or on a non-part edge. Thus identification of the open faces and non-part edges of a SDV is a prerequisite for finding prospective access positions.

4.2.1 Identification of open face and non-part edge

In the boundary of a SDV, a face that does not overlap totally/partially the any face of the part is an open face. Denote P.face and SDV[i].face the boundary face set of part P and that of the i-th sub delta volume SDV[i] respectively. The set of open faces of SDV[i] with respect to part P can be determined by the following operation

$$\text{SDV[i].open\_face} = \text{SDV[i].face} -^{*} \bigcup F \,\middle|\, F \in \text{SDV[i].face} \wedge F \bigcap \text{P.face} \neq \varnothing \qquad (4.1)$$

where SDV[i].open_face is the set of open faces of SDV[i], which is the relative complement of those faces of SDV[i] overlapping with part faces.

Here we use the intersection of *F* and P.face instead of verifying the membership of face *F* with respect to set P.face, because face *F* may be only a portion of a part face.

Theoretically, any open face can be an access face. At the present stage, let

$$\text{SDV[i].f\_acc} = \text{SDV[i].open\_face} \tag{4.2}$$

means that all open faces of a SDV are candidates of access faces of the SDV.

A pure open face is an open face that all edges of its boundary loop do not overlap with any part edge. Intuitively a face adjacent to a pure open face is also an open faces.

In the edge set of a SDV, the complement of part edges gives the non-part edge set, that is

$$\text{SDV[i].non-part\_edge} = \text{SDV[i].edge} -^{*} \bigcup E_j \tag{4.3}$$

where $E_j \in \text{SDV[i].edge} \wedge d_j = \left\| E_j \bigcap \text{P.edge} \right\| \neq \varnothing$. The second criterion guarantees that none of a portion of part edge is included in the set.

Faces situating on both sides of a non-part edge will be removed by one machining operation. Machining can start at cutting a non-part edge.

4.2.2 Access points of a general SDV

For simplification purpose, starting from this section, without causing confusions the symbol SDV instead of SDV[i] is used to represent an element of the set of sub delta volumes of a part.

As the access point of an inclined hole needs additional consideration, we call all other SDVs general SDV here.

As we know that in the boundary of a SDV the point that will be first touched by a tool is one of the following points, a non-part vertex, a point in the interior of a non-part edge or in the interior of an open face. Thus these points can be selected as prospective access points.

We define the set of access points of a given SDV, denoted SDV.p_acc, by determining all its element points. That is SDV.p_acc consists of all non-part vertices, geometry centers of open faces and that of non-part edges of the SDV. Every element of the SDV.p_acc represents a position where the cutting process to remove the SDV can be initiated.

The set of non-part vertices of a SDV, a subset of SDV.p_acc, is the complement of the vertices set of part P in the vertex set of the SDV. Denote SDV.vertex_p the intersection of vertex sets of a SDV and that of the given part,

$$\text{SDV.vertex\_p} = \text{SDV.vertex} \bigcap \text{P.vertex} \tag{4.4}$$

then we have

$$\text{SDV.vertex\_np} = \text{SDV.vertex} - \text{SDV.vertex\_p} \qquad (4.5)$$

where SDV.vertex and SDV.vertex_np denote the vertex set and the non-part vertex set of a SDV respectively.

The point set of geometrical centers of open faces, another subset of SDV.p_acc, is determined by

$$\text{SDV.geoc\_openf} = get\_geocenter(\text{SDV.open\_face}) \qquad (4.6)$$

similarly, we have the set of geometrical center of non-part edges

$$\text{SDV.geoc\_np-edge} = get\_geocenter(\text{SDV.non-part\_edge}) \qquad (4.7)$$

where SDV.geoc_openf and SDV.geoc_np-edge denote sets of geometrical centers of open faces and those of non-part edges of a SDV respectively, $get\_geocentre(\text{x})$ is the function that outputs the geometrical center of input parameter x. In equations (4.6) and (4.7) the inputs are the set of open faces and the set of non-part edges of a SDV respectively.

As defined above, the union of the sets defined by equations (4.5) through (4.7) is the set of access points of a SDV.

$$\text{SDV.p\_acc} = \text{SDV.vertex\_np} \cup \text{SDV.geoc\_openf} \cup \text{SDV.geoc\_np-edge} \qquad (4.8)$$

Vertices of a non-part edge can be added to the set SDV.p_acc even when they are part vertices. In this situation a radial complement must be made so that the first cutting point is between the two vertices if one of them is a part vertex.

To include all interior points of an open face and those of non-part edges in the set of access points, we need to replace the sets of the geometric centers in equation (4.8) by those of the interior points.

$$\text{SDV.interp\_np-edge} = \bigcup (v1, v2)_i \qquad (4.9)$$

where $(v1, v2)_i$ is the set of points in the interior of the *i*-th open edge (non-part edge) of the SDV. Points in set $(v1, v2)_i$ can be given in parameter format. For example, if the curve of non-part edge e*i* is defined by parameter function s*i*(t), $t \in [t_1, t_2]_i$. The vertices v1 and v2 are represented by s$_i$(t$_1$) and s$_i$(t$_2$) respectively. All interior points of e$_i$ are then determined as s$_i$(t), $t \in (t_1, t_2)_i$. Similarly the interior of an open face can be defined, but for saving space details will not be discussed here.

Most practically, a point chosen as an access point can be included in one of the following three categories: the midpoint of an edge, the center point of a face and the vertex of an edge. Thus replace SDV.geoc_np-edge in equation (4.8) by term

$$\mathrm{p\_acc\_np-edge} \;=\; \mathrm{SDV.vert\_np-edge} \bigcup \mathrm{SDV.geoc\_np-edge} \qquad (4.10)$$

we have

$$\mathrm{SDV.p\_acc} = \mathrm{SDV.vertex\_np} \bigcup \mathrm{SDV.geoc\_openf} \bigcup \mathrm{p\_acc\_np\text{-}edge} \qquad (4.11)$$

set SDV.p_acc defined above is the set of points consisted of points that are most often used as access points to the SDV.

4.2.3 Access point of hole SDV

Holes as one category of SDV need pay special attention when their access points are to be determined. As described in section 3.3.8, SDVs of holes (through or blind) and pockets can be included in island SDV. To distinguish a hole SDV from other island SDVs, we can use the function DisplayName, which returns the name of referenced item, to the boundary faces of an island SDV. If CylindricalFace is returned as one of the results, then the SDV is a hole. Geometrical approaches for identifying hole SDV can also be developed, to save space it will not be discussed in this dissertation.

To machine a hole, the tool axis has to be initially oriented coincidently with the hole axis. The orientation of the hole axis can be determined by applying function GetDirection to the axis of the cylindrical face, which returns a unit vector $\mathbf{s}$ on the axis. The positive direction of $\mathbf{s}$ is either corresponding to or invers to that of system axes.

The open face of a hole SDV which is selected as the access face has the following features. The face is intersected by the hole axis and the direction of its normal vector $\mathbf{n}$ is corresponding to vector $\mathbf{s}$. If two open faces are found intersecting the axis, the one whose center is farther away from the system origin is the selected face. Denote $\mathbf{m}$ the direction vector of the hole axis, then

$$\mathbf{m} = \begin{cases} \mathbf{s}, \text{ if } \mathbf{sn} \geq 0 \\ -\mathbf{s}, \text{ if } \mathbf{sn} < 0 \end{cases} \qquad (4.12)$$

The following algorithm is developed to determine the access point of a hole.

Algorithm 4.1 Determine the position of access point of a hole SDV

```
    Given a hole SDV, its axis direction vector $\mathbf{m}$ and the cylindrical face CF
    For each open face $f$ of the hole SDV {
        If $n\mathbf{m} = 1$ {
    //$f \perp \mathbf{m}$, where $\mathbf{n}$ is the unit normal vector of $f$
            SDV.p_acc = $f$.centre
      Else
            A = extreme($f$.loop, $\mathbf{m}$)
//$f$.loop is the boundary loop of $f$
            Plane_Q = make_plane(A, $\mathbf{m}$)
            SDV.p_acc = intersection(hole_axis, Plane_Q)
      }
    }
END
```

Algorithm 4.1 determines only the position of the point accessing to a hole, but it does not guaranty that the point is on the boundary of specific SDV or in the interior of previously defined stock. Thus it may happen that within a predefined stock there is no enough space for creating a planar face where hole operation can start. If so the stock as well as related SDVs has to be redefined, so that to have access point physically situated on the boundary of a corresponding SDV. In the following sections redefinitions of SDVs and stock will be discussed.

4.3 Redefinition of stock geometry

As each cutting operation starts at the moment when one of the cutting edges of a tool cuts a boundary face of current DV (workpiece) by breaking an open face of a SDV, thus an access point must situate at one of the boundary elements of current DV, which is also a boundary of the SDV being removed. As Algorithm 4.1 does not guaranty this condition being satisfied, the so determined access point SDV.p_acc of a hole may not be on the boundary of an SDV, it may even situate in the exterior of the stock. For example, as shown in Figure 4-1, the access point of an inclined hole SDV locates at the exterior of the hole SDV. We also know that for machining a hole, it is necessary to have an access face which is planar and perpendicular to the axis of the hole, and the face should be large enough so that the machining process can be conducted properly.

If the access point determined by Algorithm 4.1 is not the geometrical center of an open face of a hole SDV, SDV.p_acc$\neq$*f*.centre, this point also may be not in the interior of the given stock, or even when it is within the stock, there may be insufficient space within the DV for creating a planar access face to machine the hole. In these cases the hole SDV or the stock has to be extended. In this section we will discuss redefinitions of a given stock and an inclined hole SDV.

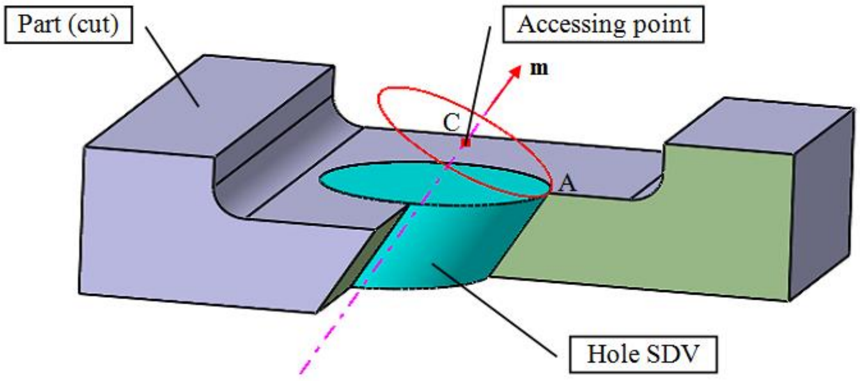


Figure 4-1 Access point of an incline hole

4.3.1 Expansion vector

To determine whether a given stock needs to be extended, and how (in what direction and to what distance) to expand it, we have to know how large the planar access face should be as measured along the coordinate axes. For expression convenience we introduce two vectors $\mathbf{r}_{D0}$ and $\mathbf{r}_{C}$

which have their end vertex at the original of the reference system XYZ, denote $\mathbf{r}_{D0} = \begin{bmatrix} \text{vdx0} & \text{vdy0} & \text{vdz0} \end{bmatrix}^T$ the primary diagonal vector of original stock, and the position vector of point C is denoted $\mathbf{r}_C = \begin{bmatrix} \text{vcx} & \text{vcy} & \text{vcz} \end{bmatrix}^T$, where C is the access point SDV.p_acc, which is the center of the perpendicular projection of the hole in Plane_Q, as shown in Figure 4-2. Plane_Q is defined in Algorithm 4.1.

As defined in 4.3.4, the reference system has its origin situating at one of the vertices of the stock, and the three edges of the stock lie coincidently with the positive axes of the system, thus vdx0, vdy0, vdz0 > 0. A given stock with a diagonal vector $\mathbf{r}_{D0}$ determines three intervals [0, vdx0], [0, vdy0] and [0, vdz0] for x, y, z coordinates respectively. It is obvious that all components of $\mathbf{r}_C$ must within corresponding intervals, otherwise the stock needs to be expanded.

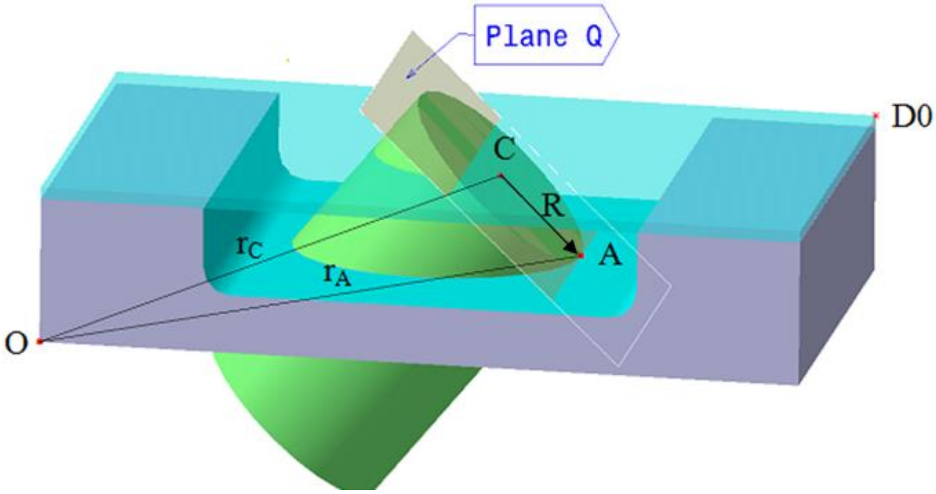


Figure 4-2 Position vectors of specific points

Before further discussing, we define two functions

$$\text{sign}(x) = \begin{cases} 1, x > 0 \\ 0, \; x = 0 \\ -1, \; x < 0 \end{cases}$$

$$zero(x) = \begin{cases} 1, x = 0 \\ 0, x \neq 0 \end{cases} \tag{4.13}$$

We also define three direction unit vectors coincident with the reference axes respectively

$$\begin{cases} \mathbf{k}_1 = \left[sign\left(\mathbf{um}\right) + zero(\mathbf{um}) \dfrac{2\mathrm{vcx} - \mathrm{vdx0}}{\left|2\mathrm{vcx} - \mathrm{vdx0}\right|} \right] \mathbf{u} \\ \mathbf{k}_2 = \left[sign\left(\mathbf{vm}\right) + zero(\mathbf{vm}) \dfrac{2\mathrm{vcy} - \mathrm{vdy0}}{\left|2\mathrm{vcy} - \mathrm{vdy0}\right|} \right] \mathbf{v} \\ \mathbf{k}_3 = \left[sign\left(\mathbf{wm}\right) + zero(\mathbf{wm}) \dfrac{2\mathrm{vcz} - \mathrm{vdz0}}{\left|2\mathrm{vcz} - \mathrm{vdz0}\right|} \right] \mathbf{w} \end{cases} \tag{4.14}$$

Where **u**, **v**, **w** are standard unit vectors of coordinate axes X, Y and Z respectively, **m** is the unit vector on the hole axis as defined in equation (4.12). Unit vectors $\mathbf{k}_1$, $\mathbf{k}_2$, $\mathbf{k}_3$ are parallel to **u**, **v**, **w** respectively. Their positive directions are determined according to both the direction of vector **m** and the position of C as relative to the symmetric planes of the stock. For example, when **m** does not perpendicular to **u**, $\mathbf{u{\bullet}m} \neq 0$, $\mathbf{k}_1$ has the same direction as **u** so long as $\mathbf{u{\bullet}m} > 0$, or as $\mathbf{u{\bullet}m} = 0$ but point C situates on the right side of the center of the stock as measured along axis-X. Otherwise $\mathbf{k}_1$ has the direction that is reversed to **u.** Similar descriptions can be made to $\mathbf{k}_2$ and $\mathbf{k}_3$.

The extremum point A used to determine Plane_Q in Algorithm 4.1 can be represented by its position vector $\mathbf{r}_A = \left[x_A \quad y_A \quad z_A\right]^T$. As point A is in the cylindrical face of the hole, and both A and C are in Plane_Q which is perpendicular to the hole axis, thus vector **R** defined by

$$\mathbf{R} = \mathbf{r}_{\mathrm{A}} - \mathbf{r}_{\mathrm{C}} \tag{4.15}$$

determines one of the radii of the hole in Plane_Q, as shown in Figure 4-3.

In Plane_Q create a circle CC of radius Rm $= \left\|\gamma \mathbf{R}\right\|$ centered at C, where $0 < \gamma \leq 1$. The value of factor γ is chosen with experience. In Figure 4-3 circle CC is drawn with the value of γ equals to 0.6. Denote $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$ the position vectors of extrema of CC in direction $\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3$ respectively.

$$\begin{cases} \mathbf{a} = extrem(\mathrm{CC},\ \mathbf{k}_1) \\ \mathbf{b} = extrem(\mathrm{CC},\ \mathbf{k}_2) \\ \mathbf{c} = extrem(\mathrm{CC},\ \mathbf{k}_3) \end{cases} \tag{4.16}$$

To create a planar face large enough so that the area enclosed by CC can be in the interior of the space defined by renewed stock (final stock), terminal vertices of all these extrema position vectors must be in the interior or on the boundary of the final stock. As vector **a** is the position vector of the extremum of CC along x-axis, it has the maximum or minimum x-component among those of the three extrema position vectors. Similarly, vector **b** and **c** have the maximum or minimum y- and z-components respectively.

When vector **a** exceeds the x-interval determined by the primary diagonal of the original stock, that is $\mathbf{au} \notin [0,\ \mathrm{vdx0}]$, then the initial stock needs to be expanded along x-axis in the direction of $\mathbf{k}_1$. Corresponding conclusions can also be drawn to vector **b** and **c** according to y-interval and z-interval respectively. Thus to judge whether the initial stock needs to be extended or not, we need only to verify one specific component of each of the three extreme position vectors, that is the x-component of **a**, the y-component of **b** and the z-component of **c** respectively. Only these specific components have significant meanings to determination of the expansion vector.

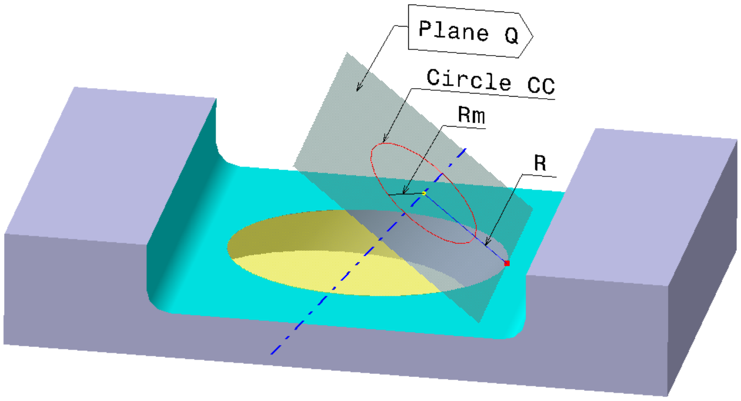


Figure 4-3 Circle CC and extended hole SDV

Denote **E** the expansion vector, its three components Ex, Ey and Ez represent the results of the dot productions of vectors **a**, **b**, **c** and standard unit axis vectors **u**, **v**, **w** respectively.

That is $\mathbf{E} = [\mathrm{Ex} \quad \mathrm{Ey} \quad \mathrm{Ez}]^T$,

$$\begin{cases} \mathrm{Ex} = \mathbf{au} \\ \mathrm{Ey} = \mathbf{bv} \\ \mathrm{Ez} = \mathbf{cw} \end{cases} \tag{4.17}$$

Denote U the diagonal matrix with standard unit vectors as its main diagonal entries.

$$\mathbf{U}=\begin{pmatrix}\mathbf{u} & & 0\\ & \mathbf{v} & \\ 0 & & \mathbf{w}\end{pmatrix}$$

Then we can arrange equation (4.17) in matrix form

$$\mathbf{E}^T=\begin{bmatrix}\mathrm{Ex} & \mathrm{Ey} & \mathrm{Ez}\end{bmatrix}=\begin{bmatrix}\mathbf{a} & \mathbf{b} & \mathbf{c}\end{bmatrix}\mathbf{U} \qquad (4.18)$$

In next section we will discuss how will an original stock be expanded according to vector **E**.

4.3.2 Geometry of final stock (SF) – single expansion vector

If there is only one inclined hole SDV in the set of SDVs of the part to be machined then the stock may need to be expanded only once. Before further discussion we define another two functions

$$pos(x)=\begin{cases}1, & if\ \ x\geq 0\\ 0, & if\ \ x<0\end{cases}$$

$$(4.19)$$

$$neg(x)=\begin{cases}1, & if\ \ x\leq 0\\ 0, & if\ \ x>0\end{cases}$$

If the expansion vector **E** determined in equation (4.18) is in the first octant (the same octant where the stock is located) then all components of **E** are positive, $\mathrm{Ex, Ey, Ez}>0$, dimensions of the final stock can be represented by its diagonal vector $\mathbf{vd}=\begin{bmatrix}\mathrm{vdx} & \mathrm{vdy} & \mathrm{vdz}\end{bmatrix}^T$, whose components are determined by

$$\begin{cases}\mathrm{vdx = vdx0} + pos(\mathrm{Ex - vdx0})(\mathrm{Ex - vdx0}),\ \mathrm{Ex}>0\\ \mathrm{vdy = vdy0} + pos(\mathrm{Ey - vdy0})(\mathrm{Ey - vdy0}),\ \mathrm{Ey}>0\\ \mathrm{vdz = vdz0} + pos(\mathrm{Ez - vdz0})(\mathrm{Ez - vdz0}),\ \mathrm{Ez}>0\end{cases} \qquad (4.20)$$

Substitute the coordinate components of the vertex vectors of the initial stock, whichever equals to vdx0, vdy0 or vdz0 correspondingly, with vdx, vdy and vdz respectively, we get a newly defined stock S_new. In the new stock, all vertices whose x-coordinates have initially the value of vdx0 are changed to vdx, similar changes are made to y- and z-coordinates whose values are vdy0 and vdz0 respectively. The initial stock remains unchanged if no component of **E** exceeding corresponding interval, in this case all factors represented as $pos(\ )$ in equation (4.20) are zero. If the stock is extended, one or more of the existing SDVs have to be redefined. Approaches for redefinition of related SDV will be discussed in next section.

Let $\Delta\mathbf{D} = \mathbf{E} - \mathbf{r}_{D0}$, where $\mathbf{r}_{D0} = [\text{vdx0} \quad \text{vdy0} \quad \text{vdz0}]^T$ is the primary diagonal vector of the initial stock, and rewrite equation (4.20) in matrix form

$$\mathbf{vd} = \mathbf{r}_{D0} + \begin{pmatrix} pos(\mathbf{u}\Delta\mathbf{D}) & & 0 \\ & pos(\mathbf{v}\Delta\mathbf{D}) & \\ 0 & & pos(\mathbf{w}\Delta\mathbf{D}) \end{pmatrix} \Delta\mathbf{D} \qquad (4.21)$$

Equations (4.20) and (4.21) give the diagonal vector of the extended stock if the expansion vector is in the first octant. However if any components of **E** exceed the left bounder of corresponding interval, it is no longer in the first octant, the given stock needs expanding in negative direction along corresponding axis. In this case the expansion of the stock cannot be simply represented by extending the diagonal.

In fact each component of **E** exceeding corresponding interval has impact on four vertices of a given stock. Instead of determining the diagonal vector, which is the position vector of one vertex, we need an equation for multiple vertices.

As we know that a stock is uniquely determined when all its vertices are defined. Given an initial stock, denote VP0 = (v01, v02, …, v08) its vertex set, denote VP = (v1, v2, …, v8) the set of vertices of the stock after expanding. Denote $\mathbf{r}_{i0} = [x_{i0} \quad y_{i0} \quad z_{i0}]^\mathrm{T}$ and $\mathbf{r}_i = [x_i \quad y_i \quad z_i]^\mathrm{T}$ the position vectors of the *i*-th vertex in sets VP0 and VP respectively. Let $\Delta\mathbf{r}_i = [\Delta x_i \ \ \Delta y_i \ \ \Delta z_i]^\mathrm{T} = \mathbf{E} - \mathbf{r}_{i0}$, we have

$$\begin{aligned} \Delta x_i &= \mathrm{Ex} - x_{i0} \\ \Delta y_i &= \mathrm{Ey} - y_{i0} \\ \Delta z_i &= \mathrm{Ez} - z_{i0} \end{aligned} \qquad (4.22)$$

For any given expansion vector $\mathbf{E} = [\mathrm{Ex} \quad \mathrm{Ey} \quad \mathrm{Ez}]^\mathrm{T}$, the components of $\mathbf{r}_i$ can be determined as following

$$\begin{cases} x_{\mathrm{i}} = zero\left(x_{i0}\right)neg\left(\mathrm{Ex}\right)\mathrm{Ex} + pos\left(x_{i0}\right)\left[pos\left(\Delta x_{\mathrm{i}}\right)\mathrm{Ex} + neg\left(\Delta x_{\mathrm{i}}\right)x_{\mathrm{i}0}\right] \\ y_{\mathrm{i}} = zero\left(y_{i0}\right)neg\left(\mathrm{Ey}\right)\mathrm{Ey} + pos\left(y_{i0}\right)\left[pos\left(\Delta y_{\mathrm{i}}\right)\mathrm{Ey} + neg\left(\Delta y_{\mathrm{i}}\right)y_{\mathrm{i}0}\right] \\ z_{\mathrm{i}} = zero\left(z_{i0}\right)neg\left(\mathrm{Ez}\right)\mathrm{Ez} + pos\left(z_{i0}\right)\left[pos\left(\Delta z_{\mathrm{i}}\right)\mathrm{Ez} + neg\left(\Delta z_{\mathrm{i}}\right)z_{\mathrm{i}0}\right] \end{cases} \qquad (4.23)$$

Equation (4.23) implies three basic rules.

1) Stock needs to be expanded when at least one component of the expansion vector exceeds corresponding interval.

2) The two opposite directions along any coordinate axis are mutually exclusive to one expansion, because each component of an expansion vector **E** can exceed either the left end or right end of corresponding interval, but not both. Each expansion vector requires the stock to be expanded in either negative or positive direction of each dimension, but not in both directions.

3) In stock expansion, faces in coordinate system planes may be offset only in negative directions of corresponding coordinate axis, while the others may be offset only in positive directions.

To give the matrix form of equation (4.23) we define two diagonal matrices

$$\mathbf{H}_{i0} = \begin{pmatrix} x_{i0} & & 0 \\ & y_{i0} & \\ 0 & & z_{i0} \end{pmatrix} \qquad (4.24)$$

$$\varepsilon = \begin{pmatrix} \mathrm{Ex} & & 0 \\ & \mathrm{Ey} & \\ 0 & & \mathrm{Ez} \end{pmatrix} \qquad (4.25)$$

$\mathbf{H}_{i0}$ is named position matrix of the *i*-th vertex of the original stock, and ε is the expansion matrix associated with a specific expansion vector **E**. Remember $\mathbf{r}_{i0} = \left[x_{i0} \quad y_{i0} \quad z_{i0}\right]^T$ and $\mathbf{r}_{\mathbf{i}} = \left[x_i \quad y_i \quad z_i\right]^T$ represent the position vectors of the *i*-th vertex of original stock and that of the final stock respectively. Equation (4.23) can then be written in more compact form

$$\mathbf{r}_i = zero\left(\mathbf{H}_{i0}\right)\left[neg(\varepsilon)\mathbf{E}\right] + pos\left(\mathbf{H}_{i0}\right)\left[\varepsilon\, pos\left(\Delta\mathbf{r}_{\mathbf{i}}\right) + \mathbf{H}_{i0} neg\left(\Delta\mathbf{r}_{\mathbf{i}}\right)\right] \qquad (4.26)$$

It is prescribed that when the functions defined in (4.13) and (4.19) is applied to a vector or matrix it is applied to all elements of the vector or matrix.

Special attention should be paid to the situation that when $\mathrm{Ez} < 0$, which means that the stock has to be extended in –z direction. In this case another setup is an obligation, because the workpiece has to be turned over so that the hole can be accessible from +z direction of the machine. For the purpose of minimize the number of setup, the point causing negative Ez will be taken into account when the hole has only one access point and the normal vector of the open face is correspond to –z direction. Further discussion can be found in later section.

Figure 4-4 shows an example of the extension of initial stock in +Z direction.

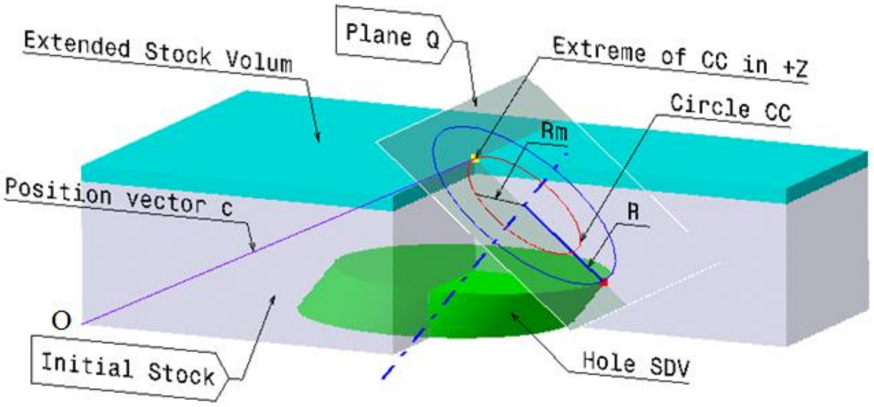


Figure 4-4 Example of the extension of initial stock

4.3.3 Geometry of final stock (SF) – multiple expansion vectors

As our approach evaluates SDVs one by one, each time when an inclined hole SDV is detected, an expansion vector is created correspondingly. Suppose totally J numbers of expansion vectors are created corresponding to J inclined hole SDVs, denote the set of expansion vectors $\mathcal{E}$

$$\mathcal{E} = \{\mathbf{E}_j\} \Big| \mathbf{E}_j = \left[\mathrm{Ex}_j \quad \mathrm{Ey}_j \quad \mathrm{Ez}_j\right]^{\mathrm{T}}, j = 1,\ 2, \cdots, \mathrm{J}$$

It is no doubt that the final stock can be achieved through J steps of dimensional changes, in each step one of the elements of $\mathcal{E}$ is used to substitute $\mathbf{E}$ in equation (4.23) or (4.26). But as we know that no matter how many times will an stock be expanded, any component of the final stock vertices must be either its original or equal to the maximum (or minimum if negative) component of all elements of $\mathfrak{E}$. Thus instead of doing multiple expansions we develop an approach which gives us the final stock through maximum two steps of expansions.

In the first expansion, if it is required, the initial stock is extended to the greatest extent in directions of negative axes, and as a result an intermediate stock is obtained. Hereafter the second expansion is applied to the intermediate stock to extend it to the greatest required extents in positive directions. It is obvious that the stock remains unchanged in a specific direction if no component of all expansion vectors exceeds corresponding interval determined by the original stock.

Denote $\mathbf{EN} = \begin{bmatrix} EN_x & EN_y & EN_z \end{bmatrix}^{\mathrm{T}}$ the vector represents the position of the vertex of the final stock which is the intersection of the three faces whose normal vectors are corresponding to negative axis directions, denote $\mathbf{EP} = \begin{bmatrix} EP_x & EP_y & EP_z \end{bmatrix}^{\mathrm{T}}$ the position vector of the final stock vertex at which three positive stock faces intersect. From their definitions we know that all components of **EN** should be less or equal to zero, while those of **EP** should be positive, that is

$$\begin{aligned} &EN_x, EN_y, EN_z \le 0 \\ &EP_x, EP_y, EP_z > 0 \end{aligned} \tag{4.27}$$

The components of **EN** are the minimum of corresponding components of all **E**$_j$, that is

$$\begin{cases} EN_x = \boldsymbol{neg}\left(\min\left(\mathrm{Ex}_j\right)\right)\min\left(\mathrm{Ex}_j\right) \\ EN_y = \boldsymbol{neg}\left(\min\left(\mathrm{Ey}_j\right)\right)\min\left(\mathrm{Ey}_j\right) \\ EN_z = \boldsymbol{neg}\left(\min\left(\mathrm{Ez}_j\right)\right)\min\left(\mathrm{Ez}_j\right) \end{cases} \tag{4.28}$$

Define vector $\min\mathbf{E}$ whose components are the minimum of corresponding components of all elements in $\mathfrak{E}$

$$\min\mathbf{E} = \begin{bmatrix} \min\mathrm{E}x \\ \min\mathrm{E}y \\ \min\mathrm{E}z \end{bmatrix} = \begin{bmatrix} \min\left(\mathrm{Ex}_j\right) \\ \min\left(\mathrm{Ey}_j\right) \\ \min\left(\mathrm{Ez}_j\right) \end{bmatrix} \tag{4.29}$$

and the corresponding matrix $\varepsilon_{\min}$ is composed by putting the elements of min**E** in corresponding positions of its main diagonal, as showing in equation below.

$$\varepsilon_{\min} = \begin{pmatrix} \min(\mathrm{Ex}_j) & & 0 \\ & \min(\mathrm{Ey}_j) & \\ 0 & & \min(\mathrm{Ez}_j) \end{pmatrix} \tag{4.30}$$

Then expansion vector **EN** can be written in a compact form

$$\mathbf{EN} = neg(\varepsilon_{\min})\mathrm{min}\mathbf{E} \tag{4.31}$$

Similarly denote max**E** the vector whose components are the maximum of corresponding components of all elements in $\mathfrak{E}$

$$\mathrm{max}\mathbf{E} = \begin{bmatrix} \mathrm{maxE}x \\ \mathrm{maxE}y \\ \mathrm{maxEz} \end{bmatrix} = \left[\max\left(\mathrm{Ex}_j\right) \quad \max\left(\mathrm{Ey}_j\right) \quad \max\left(\mathrm{Ez}_j\right) \right]^T \tag{4.32}$$

Vector **EP** has the same component of max**E** if the component exceeds the right end of corresponding interval determined by $\mathbf{r}_{D0}$, otherwise it takes the value of corresponding component of $\mathbf{r}_{D0}$. Denote $\boldsymbol{\delta}$ the difference between max**E** and $\mathbf{r}_{D0}$,

$$\boldsymbol{\delta} = \mathrm{max}\mathbf{E} - \mathbf{r}_{D0} = \left[\delta x \quad \delta y \quad \delta z\right]^T$$

The components of **EP** are determined as following

$$\begin{cases} EP_x = pos\left(\delta x\right)\mathrm{maxE}x + pos\left(\mathrm{maxE}x\right)neg\left(\delta x\right)\mathrm{vdx0} \\ EP_y = pos\left(\delta y\right)\mathrm{maxE}y + pos\left(\mathrm{maxE}y\right)neg\left(\delta y\right)\mathrm{vdy0} \\ EP_z = pos\left(\delta z\right)\mathrm{maxEz} + pos\left(\mathrm{maxEz}\right)neg\left(\delta z\right)\mathrm{vdz0} \end{cases} \tag{4.33}$$

Define diagonal matrices $\varepsilon_{\max}$ and $\boldsymbol{\Lambda}$ with components of max**E** and $\boldsymbol{\delta}$ as their elements in the main diagonals respectively,

$$\varepsilon_{\max} = \begin{pmatrix} \mathrm{maxE}x & & 0 \\ & \mathrm{maxE}y & \\ 0 & & \mathrm{maxEz} \end{pmatrix} \tag{4.34}$$

$$\Lambda = \begin{pmatrix} \delta x & & 0 \\ & \delta y & \\ 0 & & \delta z \end{pmatrix} \tag{4.35}$$

and rewrite equation (4.33) in matrix form

$$\mathbf{EP} = \varepsilon_{\max} pos(\boldsymbol{\delta}) + pos(\varepsilon_{\max}) neg(\boldsymbol{\Lambda}) \mathbf{r}_{D0} \tag{4.36}$$

Denote $\mathbf{g}_i = [gx_i \quad gy_i \quad gz_i]^{\mathrm{T}}$ the position vector of the *i*-th vertex of the intermediate stock after expanding with **EN,** and denote $\mathbf{f}_i = [fx_i \quad fy_i \quad fz_i]^{\mathrm{T}}$ the position vector of the *i*-th vertex of the final stock. Similar to matrix $\mathbf{H}_{i0}$ defined in equation (4.24), we define two diagonal matrices $\mathbf{H}_i$ and $\mathbf{MP}$ corresponding to $\mathbf{g}_i$ and **EP** respectively as following

$$\mathbf{H}_i = \begin{pmatrix} gx_i & & 0 \\ & gy_i & \\ 0 & & gz_i \end{pmatrix}, \mathbf{MP} = \begin{pmatrix} EPx & & 0 \\ & EPy & \\ 0 & & EPz \end{pmatrix} \tag{4.37}$$

By expanding the original stock along negative axes to the extent of **EN** we get an intermediate stock, whose vertices are represented by their position vectors $\mathbf{g}_i$ determined as the following

$$\mathbf{g}_i = zero(\mathbf{H}_{i0}) \mathbf{EN} + pos(\mathbf{H}_{i0}) \mathbf{r}_{i0} \tag{4.38}$$

Hereafter the second expansion which turns the intermediate stock into the final stock can be conducted to the extent of **EP** along positive axes. The vertices of the final stock are defined as following

$$\mathbf{f}_i = neg(\mathbf{H}_i) \mathbf{g}_i + pos(\mathbf{H}_i) \left[\mathbf{MP} pos(\Delta \mathbf{f_i}) + \mathbf{H}_i neg(\Delta \mathbf{f_i}) \right] \tag{4.39}$$

where $\Delta \mathbf{f}_i = \mathbf{EP} - \mathbf{g}_i$. The final stock is denote as *SF*.

In practice a physical material stock should be prepared with zero upper deviations in all dimensions as to the model defined in equations (4.39).

4.4 SDVs Updating

As we know that the open face of an inclined hole SDV obtained by delta volume decomposition may be not orthogonal to the hole axis, in order to create a planar access face for an inclined hole, we should first know if there is enough space within the original delta volume, if there is not then we need also know how to expand the stock dimensions to make it possible to create such a face. These topics have been discussed in section 4.3. When enough space in the delta volume is available for creating an access face for an inclined hole SDV, no matter the stock has been in fact expanded or not, the next question is how the redefine the SDVs of inclined holes, what impacts does this redefinition have on other SDVs, and how to define the SDVs correspond to increased volume of the stock if there is any expansion. These topics will be discussed in this section.

4.4.1 Redefinition of inclined hole SDV

The final stock defined in section 4.3.2 or 4.3.3 encloses major portion of the circle CC in its interior, thus by extending the original hole SDV to Plane_Q we can obtain a new hole SDV whose boundary face in Plane_Q satisfies the prerequisite of an access face to a hole. There are several different geometric ways to realize this extension. Here we introduce an approach to expand the hole SDV by means of extruding and trimming.

Suppose SDV[i] is the original sub delta volume of an inclined hole, F is one of its open faces where extremum point A is determined. Extrude F along the hole axis in the direction outwards the volume. The length of the extrusion equals to the farthest distance between Plane_Q and the loop of F. Trim the extruded volume with Plane_Q, and also with boundary face(s) of the final stock if necessary. The extruded volume needs to be trimmed with a stock face only if it intersects the stock face and the area of the intersection is not zero, see Figure 4-5. To save space the algorithm for trimming the extrusion will not be developed in this work. As a result we get an extended volume of the hole SDV[i], denoted SDV[i]_ext, which is within the interior of the final stock. Combine SDV[i]_ext with the original SDV[i], we get a redefined hole SDV denoted SDV[i]_rdf

$$\text{SDV[i]\_rdf} = \text{SDV[i]} \cup \text{SDV[i]\_ext} \qquad (4.40)$$

Arrange SDV[i]_rdf in the set of updated SDVs denoted SDVRDF, suppose its index number is I, that is

$$\text{SDVRDF[I]} = \text{SDV[i]\_rdf} \tag{4.41}$$

A link between SDV[i] and SDVRDF[I] has to be created so that one of them can be cited by the other correspondingly. As the index number is unique to an element in a set, thus we can assign the same number i to the index of SDVRDF[I], that is let I = i, then it is easy to find corresponding elements in both sets of SDVs and SDVRDF.

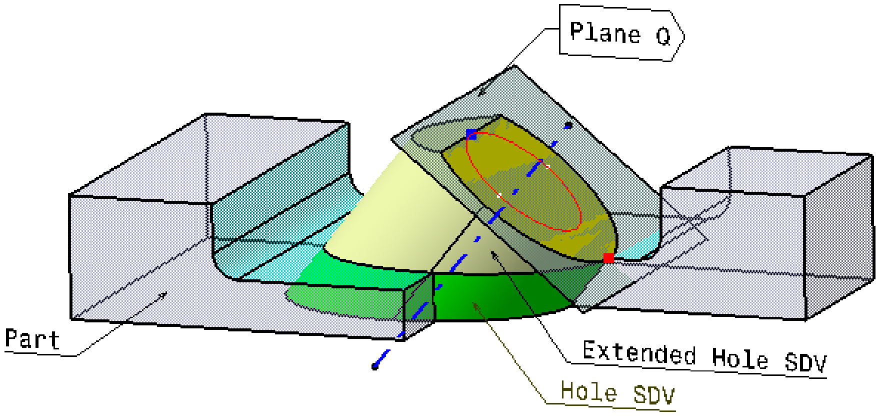


Figure 4-5 Redefinition of the inclined hole SDV

In practice it is not necessary to trim the extrusion with stock face(s), because either it is trimmed or not machining parameters for producing the hole will be the same. But a trimmed extrusion shows precisely the volume of material to be removed.

4.4.2 Final periphery SDV

A SDV having boundary face(s) overlapping with at least one of the stock boundary faces is defined as a periphery SDV. After dimension extension of the stock corresponding changes to certain periphery SDVs have to be made so that they will be bounded by the final stock face(s).

Examine the expansion vector **E**, **EN** or **EP** defined in section 4.3.1 through 4.3.3, identify their component(s) that exceeds corresponding interval(s) determined by $\mathbf{r}_{D0}$, then we know in which direction the stock is expanded, in other words we know where a volume is added to the original stock. The union of original periphery SDV and corresponded added volume will take place of the original periphery SDV and become one element of the final periphery SDVs.

If $\mathrm{Ex} < 0$ or $\mathrm{ENx} < 0$, the stock has been extended in negative x-axis, slit the final stock with plane x = 0, the portion in the negative half space of the plane is defined as the final periphery SDV in the half space x ≤ 0, denoted SDV_nx. To simplify our expressions we define a special solid which is big enough to be the upper set of all volumes of removable material situating in the negative half space of x = 0, name it negative x solid and denote it NXS. Similarly we define negative y solid and negative z solid denoted NYS and NZS respectively. With the help of these special solids, we can determine all periphery SDVs in negative half spaces x ≤ 0, y ≤ 0, and z ≤ 0

$$\begin{cases} \mathrm{SDV\_nx} = SF \bigcap \mathrm{NXS} \\ \mathrm{SDV\_ny} = SF \bigcap \mathrm{NYS} -^{*} \mathrm{SDV\_nx} \\ \mathrm{SDV\_nz} = SF \bigcap \mathrm{NZS} -^{*} (\mathrm{SDV\_nx} \bigcup \mathrm{SDV\_ny}) \end{cases} \qquad (4.42)$$

where SDV_ny and SDV_nz denote periphery SDVs in the half spaces y ≤ 0 and z ≤ 0 respectively. From their definitions in equation (4.42) we know that these three periphery SDVs do not have intersection with each other, thus double machining of any portion of these volumes is avoided.

To determine periphery SDVs in three positive half spaces of the planes passing the terminal vertex of $\mathbf{r}_{D0}$, define three planes PL1: z – vdz0 = 0, PL2: y – vdy0 = 0 and PL3: x – vdx0 = 0, all of them pass the terminal vertex of $\mathbf{r}_{D0}$ and are parallel to corresponding coordinate planes respectively. In the positive half space of PL1, z – vdz0 $>$0, define positive z solid PZS, and in that of PL2 and PL3 define positive y solids PYS and positive x solid PXS respectively. These positive solids are big enough to have the volumes of removable material situating in the positive half space of PL1, PL2 and PL3 as their subset volume respectively. Similar to equation (4.42) we define the periphery SDVs in positive domains

$$\begin{cases} \mathrm{SDV\_pz} = SF \bigcap \mathrm{PZS},\ \mathit{if}\ \ \mathrm{Ez\ or\ EPz} > \mathrm{vdz0} \\ \mathrm{SDV\_py} = SF \bigcap \mathrm{PYS} -^{*} \mathrm{SDV\_pz},\ \mathit{if}\ \ \mathrm{Ey\ or\ EPy} > \mathrm{vdy0} \\ \mathrm{SDV\_px} = SF \bigcap \mathrm{PXS} -^{*} (\mathrm{SDV\_pz} \bigcup \mathrm{SDV\_py}),\ \mathit{if}\ \ \mathrm{Ex\ or\ EPx} > \mathrm{vdx0} \end{cases} \qquad (4.43)$$

A special solid can be any form, for example a hemisphere, a cube or a rectangular volume. A hemisphere solid is highly recommended. The radius of the sphere, for example, can be the maximum of axis move distance of a machine in corresponding dimension. No matter what shape does a special solid has, it can be defined in such a way that it is symmetric about corresponding plane of symmetry of the final stock. Figure 4-6 shows the hemisphere centered at the geometric center of the top face of initial stock. The extended stock volume is determined as the intersection of the final stock and the hemisphere.

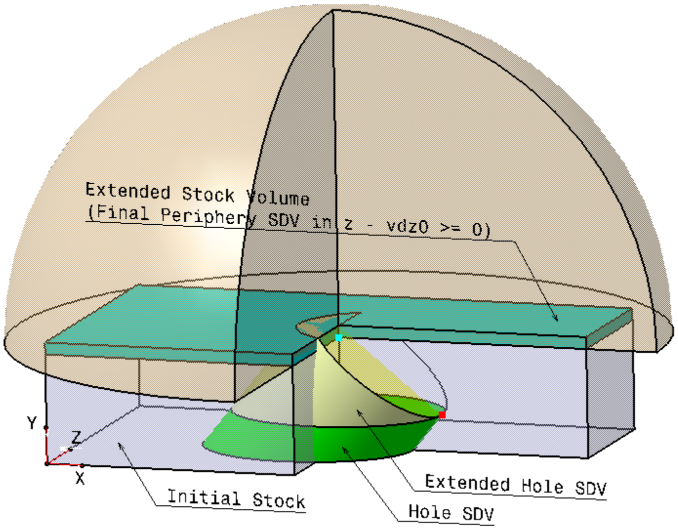


Figure 4-6 Example of defining final periphery SDV

Newly defined periphery SDV(s) are grouped in sets NSDV and PSDV as they are in negative and positive half space of corresponding planes respectively. It is intuitive that no facing machining should be applied to any newly added periphery SDV because within the volumes of these SDVs there exist the planar access faces as well as the extended volumes of the inclined hole SDVs. In practice if facing process is necessary additional volume should be added.

After expanding a stock in the negative direction of an axis the corresponding stock face does not locate in the coordinate plane any more. The stock vertex which was originally positioned at the origin of coordinate system has its new position. If the same reference system is to be used in setup, correct information about the final positions of stock faces is very important. Highlighting the position(s) of these face(s) will be helpful to avoid mistakes in setup.

4.4.3 Subdivision of SDVs intersected by FCC

For the purpose of creating a planar face as the access face of an inclined hole SDV, all existing SDVs that intersect with the extended volume of the hole have to be redefined. In section 4.3.1 we have defined the circle CC in Plane_Q. The area bounded by CC, denoted as FCC, represents the planar access face necessary to machining the hole. If we redefine the sub delta volumes that intersect FCC in such a way that each of them has a boundary face overlapping with Plane_Q, then by removing the portion volumes of these SDVs located in the positive half space of Plane_Q, a planar face enclosing FCC can be physically created.

Suppose SDV[j] is such a sub delta volume that is adjacent to an original inclined hole SDV[i], the intersection of SDV[j] and the hole SDV[i] is one of the open face of SDV[i] where the extremum point A is detected. That is

$$\mathrm{F} = \mathrm{SDV[i]} \cap \mathrm{SDV[j]} \tag{4.44}$$

If SDV[j] is an upper bounder of face FCC, or in other words face FCC is completely enclosed within the volume of SDV[j], that is

$$\mathrm{FCC} = \mathrm{FCC} \cap \mathrm{SDV[j]} \tag{4.45}$$

then SDV[j] is the sole SDV that needs to be redefined for creating a the planar face overlapping FCC.

When equation (4.45) is satisfied, redefinition of SDV[j] can be achieved by slitting SDV[j] with Plane_Q, thus results two smaller sub delta volumes, SDV_a[j] and SDV_b[j]. The one, say SDV_a[j], situated in the positive half space of Plane_Q is named as positive portion of SDV[j] which should be removed before machining the hole. The other named as the negative portion of SDV[j] can be removed in any machining step succeeding removal of the hole.

As the volume of removable material represented by SDV[j] is now defined by SDV_a[j] and SDV_b[j], corresponding machining operation should also be determined on the base of these two sub delta volumes. For identification purpose, we create other two sets of sub delta volumes, denoted SDVA and SDVB, consisting of respectively positive and negative portion(s) of SDVs that intersect the expanding volumes of inclined holes.

In addition to SDV[j] if there are other SDVs, including periphery SDV(s), intersecting with face FCC, their positive as well as negative portions can be determined similarly.

Associations between the inclined hole SDVs and the positive as well as the negative portions of corresponding SDVs can be made by creating a set of index citations. Denote Idxcitation( ) the array of index citation, which has seven dimensions representing, in the order,

1) idx_inclind, the index of inclined hole as that in the set of SDVs;
2) Nbr_pstv, the number of SDVs which have no zero volume positive portions as referred to Plane_Q corresponding to the inclined hole;
3) Nbr_ngtv, the number of SDVs which have no zero volume negative portions as referred to Plane_Q corresponding to the inclined hole;
4) idx_pstv, one dimension array, indexes of SDVs having positive portions;
5) idx_A, indexes of positive portions of SDVs in SDVA;
6) idx_ngtv, one dimension array, indexes of SDVs having negative portions;
7) idx_B, indexes of negative portions of SDVs in SDVB;

Each row of the array Idxcitation( ) consists the indexes of SDVs as well as their positive and negative portions associated with an inclined hole.

We can create temporarily a combination of all these SDVs

$$\mathrm{SDV\_comb} = \bigcup \mathrm{SDV[k]},\ \mathit{Area}(\mathrm{SDV[k]} \cap \mathrm{FCC}) \neq \varnothing \qquad (4.46)$$

Separate SDV_comb into two sub volumes with Plane_Q, SDV_comb_a located in the positive half space of Plane_Q and SDV_comb_b in the negative half space, see Figure 4-7. As the planar access face FCC can be physically created only when the volume of SDV_comb_a has been removed, thus it can be considered to remove this sub volume as a whole in one machining process if it is accessible from one direction.

SDV_comb_b can be taken either as one unit of removable volume or as several individual volumes each is the portion (in the negative half space of Plane_Q) of one of these SDV[k].

In Figure 4-7, the recombined sub delta volume SDV_comb_b is showed in the way that its intersection with the extended hole volume is removed, as the hole will be removed before. In fact the geometry of SDV_comb_b can be defined without removing the intersection volume, and allow this volume to be machined twice. Which is better? This question is left for further study.

To remove the SDV_comb_b as a whole or to remove its component volumes individually is a question of optimization. This topic is also left for further studies.

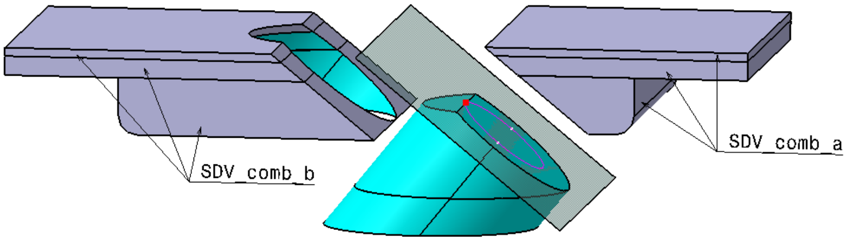


Figure 4-7 Subdivisions of SDVs

In addition to the volume representation, we need also an algorithm which determines a correct sequence of removing volumes. As mentioned above, in machining process the positive portions of all SDV[j] that intersect face FCC must be the predecessors of the expanded hole SDVRDF[i], while the negative portions must be the successors. Detail discussion is left to future research.

4.5 Tool axis orientation

In most cases, at each access point there are theoretically infinitive possible directions according to which a tool can be oriented. From the perspective of roughing, some of the most reasonable initial tool axis orientations are:

1) Parallel to Z axis
2) perpendicular to the access face or the bottom face of a SDV
3) coincident with the symmetrical axis of the volume
4) parallel to one of the column edge which shares vertex with the access face
5) coincident with the center line (the line passing the geometric centers of the top face and bottom face of a SDV)

To remove a volume of material the cutting edges of a tool (the tip or cutting side) has to be accessible to every boundary face of a SDV. The initial orientation of a tool can be optimized under several criteria, for example minimum tilt angle necessary for removing the whole SDV, maximal diameter of the tool can be used, etc. The subjects concerning optimization are left to future researches. In this section we will only discuss the determination of the initial orientation of a tool from the perspective of the accessibility to a SDV.

4.5.1 Parallel to Z axis

Parallel to Z axis is the priority direction for a tool to perform roughing operations. To verify the possibility of a SDV to be machined by a tool approaching from +Z axis, give SDV.open_face = $\{f_i\}$, if $\exists\ f_j \in \{f_i\}$ whose normal is denoted as $\mathbf{n}_j$, that

$$\mathbf{wn}_j > 0 \qquad (4.47)$$

And that

$$\text{prj}f_j \supseteq \bigcup \text{prj}f_i \,\big|\, f_i,\ f_j \in \{f_i\} \qquad (4.48)$$

then the SDV can be machined from +Z direction if the operation starts at access face f_j. In criterion (4.48) $\text{prj}f_j$ and $\text{prj}f_i$ denote the projection of faces f_j and f_i onto XY plane along direction –**w** respectively.

4.5.2 Perpendicular to access face or bottom face

Give a SDV, together with one of its access faces f_acc, one of its access points SDV.p_acc corresponding to f_acc, and plane Q which is tangent to f_acc at SDV.p_acc. Denote **n** the normal vector of f_acc at SDV.p_acc, the positive direction of **n** is outwards the volume SDV. If the projection of f_acc onto Q along **n** is an upper bound of those of all other faces of the SDV, then the given SDV is bottom narrowed (or top open) as viewed from –**n.**

Denote *SA* and *SB* the projection of f_acc and SDV.f respectively

$$\begin{aligned} SA &= \mathit{project}(\text{SDV.f\_acc},\ \text{Q}, \mathbf{n}) \\ SB &= \mathit{project}(\text{SDV.f},\ \text{Q}, \mathbf{n}) \end{aligned} \qquad (4.49)$$

If

$$SA \supseteq SB \qquad (4.50)$$

then the SDV is bottom narrowed.

Plane Q can also be chosen tangent to the bottom face f_bott of a SDV at the geometric center of f_bott. Denote p_c the geometric center of f_bott, **m** the normal of f_bott at p_c, **m** pointing outward the SDV, if projections of faces of the SDV onto Q along **m** are completely overlapped by that of f_acc, then the SDV is also bottom narrowed.

To machine a bottom narrowed SDV, a tool can to be oriented perpendicular to plane Q, see Figure 4-8. The direction coincident with the normal of the bottom face is preferably suggested. An algorithm for verifying a bottom narrowed SDV can be developed in such a way that the plane Q is first chosen related to face f_bott, and then related to f_acc if the first verification turns out failure.

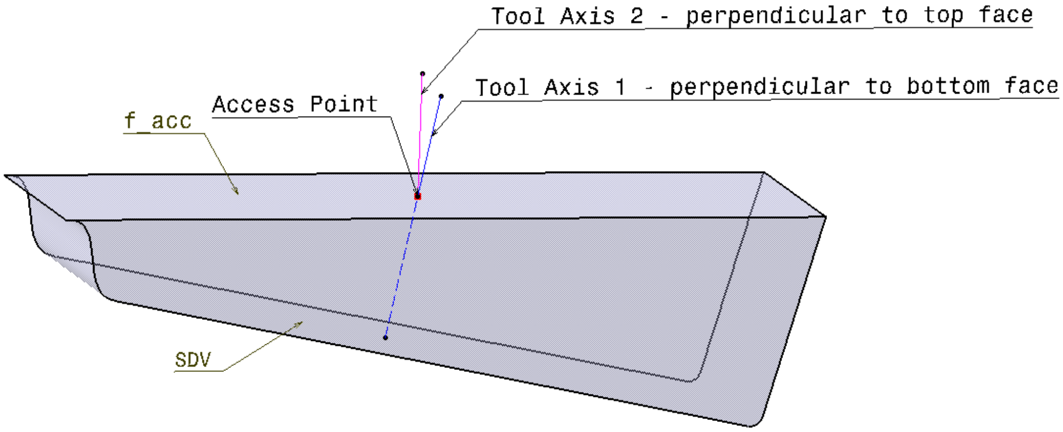

Tool Axis 2 - perpendicular to top face
Tool Axis 1 - perpendicular to bottom face
Access Point
f_acc
SDV

Figure 4-8 Tool axis orientation – perpendicular to the top/bottom face

Theoretically if all wall faces of a SDV are open faces, the half space to position the tool axis vector can be determined in such a way by making a plane passing the access point and perpendicular to the normal vector of the bottom face at its geometrical center. The positive side of the plane is corresponding to the positive direction of normal vector of the access face. Development of detail approaches for this situation is left for future research.

4.5.3 Coincident with the axis of rotating volume

In this research a rotating removable volume is classified in the category of island SDV, the profile of the rotating surface can be linear or not. When the profit is not linear, and the access face is not perpendicular to the rotation axis, Algorithm 4.2 may give the result of Tool_axis vector $\mathbf{s}=\mathbf{0}$. For a rotating volume we can simply orient the tool axis coincident with the rotation axis. Thus it is necessary to identify the rotation axis first.

The following algorithm is developed for identifying the rotation axis and determining the tool axis vector for removing a rotating SDV.

Algorithm 4.3 Tool orientation for rotating SDV

```
    Given an island SDV and its access face f_acc
    f_acc.edge = get_edge (f_acc.vertex)
    f_wall = get_face(f_acc.edge) – f_acc
    AX = get_axis(f_wall)
    if  AX = get_axis(f_wall) $\neq \varnothing$ {
//then f_wall is a rotating face
        ID_rotation = ‘true’
    else
        ID_rotation = ‘false’
//SDV is not a rotating volume
    }
    If ID_rotation = ‘true’
        Make a unit vector **r** passing the geometric center of f_acc and parallel to AX
        Tool_axis vector **s** = **r**
    Else
        Tool_axis vector **s** = **0**
End
```

The positive direction of the unit vector **r** is in accordance with the normal vector **n** of the access face at its geometric center directing outwards the volume. When the tool axis vector is determined, the rotating volume has to be checked to know whether its wall face is folded.

4.5.4 Tool axis for linear side SDV

In many cases, all side edges (column edges) of a SDV are linear, we name this kind of SDV linear side SDV. If a SDV is found not in any of the three cases mentioned in sections 4.5.1 through 4.5.3, in other words vector $\mathbf{s} = \mathbf{0}$ is the result of both Algorithm 4.2 and Algorithm 4.3 , the given SDV can be examined to know if it is a linear side SDV. To machine a linear side SDV, under the condition that it is angularly manipulable (definition will be given in the next sub section), the tool can be postured initially coincident with the center line Lcenter.

As discussed in chapter 4, to each open face of a SDV there is a corresponding bottom face. Once an open face is selected as the access face f_acc, the corresponded bottom face f_bott is determined. We define the line passing the geometrical centers of both f_acc and f_bott as the center line, denoted Lcenter, see Figure 4-9.

As we did to a hole SDV, plane Q can be chosen perpendicular to Lcenter and passing either center of f_acc or f_bott. If the projection of the bottom face onto Q is a subset of that of the open face f_acc, then the SDV is bottom narrowed. In this case the tool axis can be coincident with Lcenter, otherwise it can be oriented parallel to one of the column edges, which is one of the boundary edges of the face whose normal vector has the smallest angle with +z axis as compared to other side faces of the SDV. Further development to the algorithm of tool axis determination will not discuss here.

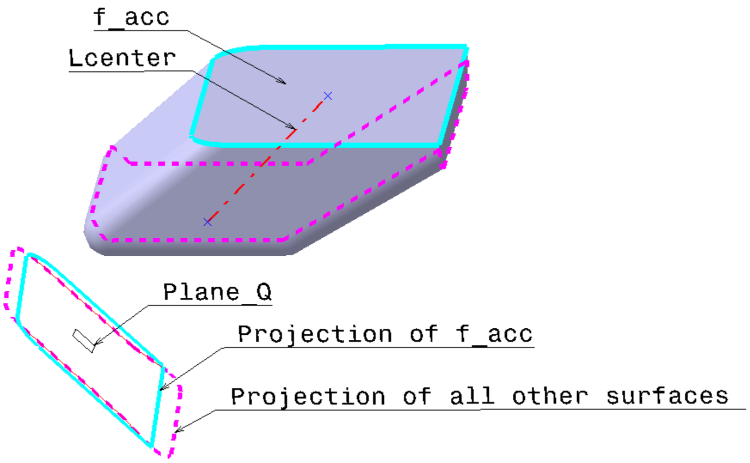


Figure 4-9 Center line and Plane_Q

One special category of linear sided SDV is named straight column SDV, whose column edges are all parallel linear segments. To remove a straight column volume of material, in the condition that the column faces are not blocked by any part face, the tool can be oriented parallel to any of the column edges. The situation that a face is blocked by another part face will be discussed later in section 4.7.

As the side faces of a column can be generated by sweeping the loop of access face along a linear segment that is parallel to one of its column edge. Thus to identify a straight column SDV, we need only to verify whether it is true that the projection of its side faces onto a plane perpendicular to one of its side edge completely overlap with that of the loop of access face. The following algorithm is based on this idea.

Algorithm 4.4 Straight column SDV identification

```
    Given a SDV and its access face f_acc
    f_acc.vertex = getvertex(f_acc)
    edge_column = getedge(f_acc.vertex) – getedge(f_acc)
    ID_straightcolumn = ‘true’
    For $e_i \in$ edge_column
        If $e_i$ is not linear
            ID_straightcolumn = ‘false’
    End for
    If ID_straightcolumn = ‘true’
        $\mathbf{R} = \frac{\mathbf{r}_1 - \mathbf{r}_2}{|\mathbf{r}_1 - \mathbf{r}_2|}$
// $\mathbf{r}_1$ and $\mathbf{r}_2$ are position vector of the initial and terminal vertices of $e_1$ respectively
        p = get_extrem(f_acc, $\mathbf{R}$)
        Q = make_plane(p, $\mathbf{R}$)
        side_face = $\bigcup$ get_leftface($e_j \mid e_j \in$ edge_column)
        proj_A = projection(side_face, Q, $\mathbf{R}$)
        proj_B = projection(f_acc.loop, Q, $\mathbf{R}$)
        if proj_A$\bigcap$proj_B = proj_A
            ID_straightcolumn = ‘true’
    Else
            ID_straightcolumn = ‘false’
    }
End of Algorithm 4.4
```

If the result exported by Algorithm 4.4 is ‘true’, the tool axis vector can be determined as either the unit vector **R** or the inverse of **R** according to the direction of **R** with respect to the normal vector of f_acc at its geometric center.

$$\mathbf{s} = \begin{cases} \mathbf{R}, & \mathbf{Rn} \geq 0 \\ -\mathbf{R}, & \mathbf{Rn} < 0 \end{cases} \qquad (4.51)$$

where **n** is the normal vector of the access face f_acc at its geometrical center.

4.5.5 Detecting folded face

In Algorithm 4.2 the open face f, or the bottom face corresponding to f, is taken as the reference face. When calculate sum_projection_n or sum_projection_m it is assumed that every wall face of the SDV is not folded with respect to the specific vector **s**. A face is said to be folded in direction **s** if there is a path in the face which is the intersection of the face and a plane parallel to **s**, along the path the normal vector of the face changes to reverse with respect to **s.** Figure 4-10 shows an example of folded face in which normal vectors N1 and N2 are opposite in directions as respect to **s**.

Intuitively, if face f(x, y, z) = 0 is folded with respect to **s**, there must exist a point p(a, b, c) in face f where the tangent plane of face f is parallel to **s**, denote ***N*** the normal vector of face f at p(a, b, c), then we have $\boldsymbol{N} \perp \mathbf{s}$. As we know the normal vector of a face at a given point equals to the gradient of the implicit function expressing the face. Thus the point p must belong to the solution set of following equation if the set is not empty

$$\mathbf{s} \cdot \boldsymbol{N} = \mathbf{s} \nabla f(x, y, z) = 0, \ (x, y, z) \in f \qquad (4.52)$$

Faces satisfying equation (4.52), except planar, cone or cylinder faces parallel to **s**, are folded faces with respect to vector **s**. The following algorithm is developed to determine folded face.

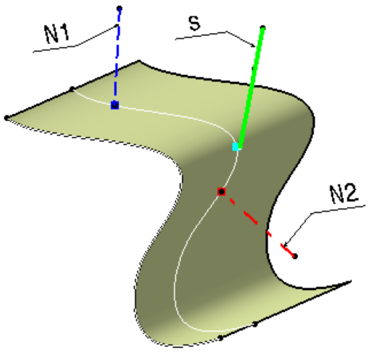


Figure 4-10 Example of folded face

Algorithm 4.5 Detecting folded face

```
Given a SDV and a specific vector **s**
For each face f in the face set SDV.face
//      except the face that **s** is one of its normal {
        f_xyz = getimpfunction(f)
//getimpfunction(f) is the process that outputs the implicit function of f
        If grad(f_xyz) = constant {
//f_xyz is a plane
                Fold_judg = “fault”
        Else
                Rs = Solve(**s**·grad(f_xyz) = 0) {
                If $Rs \neq \varnothing$
                        rs = lowest point in Rs as in direction **s**
                        $\mathbf{n} = \text{grad}(f\_xyz)|_{rs}$
                        P = make_plane(rs, **n**)
```

```
                    $X = P \bigcap f$
                    If X parallel to $\mathbf{s}$
                              Fold_judg = "fault"
                    Else
                      Fold_judg = "true"
//the face is folded
               Else
                    Fold_judg = "fault"
          }
} END of Algorithm 4.5
```

The access direction to a folded side face will be discussed in section 4.5.6.

Another situation similar to a fold face can occur to some SDVs. Instead of being one continuous face, several faces adjacent one to the other and some of them are sheltered by some others. That is to say that there are more than one faces adjacent one to the other in such a way that one is in a higher position than the other. If going down form one face to the other, the z-components of the normals of these faces change to opposite. That is, denote $\{f_i\}, i = 1,...,N$ the set of the s of SDV, and f_i is adjacent to f_{i+1}, denote $\mathbf{n}_j = (n_{xj} \ n_{yj} \ n_{zj})^T$ the normal vector of face $f_j \in \{f_i\}$ where

$$n_{zj} = \max\left(\frac{\partial f_j}{\partial z}\right)$$

without loss of generality suppose $n_{z1} \geq 0$ and $n_{zi} \geq n_{zi+1}$, if

$$\prod_{i=1}^{N} pos(n_{zi}) = 0 \text{ or } \prod_{i=1}^{N} neg(n_{zi}) = 0 \tag{4.53}$$

then $\{f_i\}, i = 1,...,m$ is a folded face chain as respect to Z.

4.5.6 Access direction to a folded side face

A bended face is a self-sheltered face which can be separated into two portions: sheltering and sheltered portions. These two portions can then be treated as two sheltering and sheltered faces respectively. The curve separating the folded face can be determined as follows.

$$\mathbf{X} = \left\{ \mathbf{x}(p) \middle| p \in f, \mathbf{n}(p)\cdot\mathbf{k} = 0, \text{ and } \left.\frac{\partial n_z}{\partial z}\right|_p > 0 \right\} \tag{4.54}$$

All points $p \in f$ whose position vectors consist the set **X** as defined in (4.54) above, succinctly the points in **X**, consist curve *C*. Split face *f* with *C*, we get two portions: the higher and lower, the higher portion is a sheltering face of the lower one, which in turn is the sheltered face. The access directions to these two portion faces can be determined by procedures discussed below.

Faces in a folded face chain can be separate into two groups by the curve C determined by equation (4.54) if one of these faces is a self-folded face, or by the common boundary edge where the two adjacent faces have opposite n_z . The tool axis orientation has to be determined according to directions of faces in both groups. Denote $\{f_i\}, i = 1,..., N$ the set of the s of SDV which form a folded face chain, and f_i is adjacent to f_{i+1}, denote $\mathbf{n}_j = (n_{xj}\ n_{yj}\ n_{zj})^T$ the normal vector of face $f_j \in \{f_i\}$. Without loss of generality suppose $n_{z1} \geq 0$ and $n_{zi} \geq n_{zi+1}$, to remove the SDV from one direction, the tool axis must be positioned in the negative half space of the plane perpendicular to $\mathbf{n}_1$, if facing +z is the positive side. In the contrary if $n_{z1} < 0$ and $n_{zi} \leq n_{zi+1}$, the tool axis must be positioned in the positive half space of the plane perpendicular to $\mathbf{n}_1$.

For a SDV having folded face chain, the criteria of being able to be machined from one direction can be expressed as following

$$\begin{gathered} \prod_{i=1}^{N} pos(n_{xi}) > 0 \text{ , and } \prod_{i=1}^{N} pos(n_{yi}) > 0 \text{ or } \prod_{i=1}^{N} neg(n_{yi}) > 0 \\ \text{or} \\ \prod_{i=1}^{N} neg(n_{xi}) > 0 \text{ , and } \prod_{i=1}^{N} pos(n_{yi}) > 0 \text{ or } \prod_{i=1}^{N} neg(n_{yi}) > 0 \end{gathered} \tag{4.55}$$

In this work we discuss the simplest situation, suppose that all common edges of adjacent faces are linear segments and all edges sharing a vertex of a common edge are not part edges. Under this circumstance, the tool axis direction is determined as follows.

If $n_{z1} < 0$ and $n_{zi} \leq n_{zi+1}$, i = 1, 2, …, N-1, let

$$n_{zM} = \max(\frac{\partial f_N}{\partial z}) \qquad (4.56)$$

where f_N is the implicit equation of face f_N. Denote P the plane perpendicular to xy plane and containing the normal $\mathbf{n}_p$ of f_N at point p where n_{zM} is determined, denote Q the tangent plane of f_N at the same point p, and denote R the intersection of P and Q,

$$R = P \bigcap Q \qquad (4.57)$$

Let $\mathbf{r}$ the unit vector on R, $r_z \geq 0$. Let

$$\mathbf{n}_p = \left. \frac{\nabla f_N}{\|\nabla f_N\|} \right|_p \qquad (4.58)$$

Then the tool axis vector $\mathbf{s}$ can be any vector that

$$\mathbf{sn}_p = 0 \text{ and } \mathbf{sr} \geq 0 \qquad (4.59)$$

If $n_{z1} > 0$ and $n_{zi} \geq n_{zi+1}$, let

$$n_{zM} = \max(\frac{\partial f_1}{\partial z}) \qquad (4.60)$$

where f_1 implicit equation of face f_1. Similarly plane P and Q can be determined passing point q where n_{zM} is detected. Denote

$$\mathbf{n}_q = \left. \frac{\nabla f_1}{\|\nabla f_1\|} \right|_q \qquad (4.61)$$

Then the tool axis vector is determined by

$$\mathbf{sn}_q = 0 \text{ and } \mathbf{sr} \geq 0 \qquad (4.62)$$

4.6 Angular operability

In sections 5.5.1 through 5.5.3 we have discussed methods for determining initial tool axis orientations for three kinds of SDVs, in fact there are some SDVs that cannot be included in these categories. Furthermore questions such as “Is a machine capable to provide required orientation to a tool?” , “Is there a direction from which all part faces of given SDV are accessible?” have not been answered. Here we define angular operability as the capability that a machine can provide to orient a tool in postures to access part faces of given SDV as required. In this section we are going to discuss the representation of the domain that a machine can provide to pivot a tool, required angular posture and the angular operability of a part.

Among factors that restrict the accessibility of faces of a SDV, the geometries of the SDV and the angular deflection capacity of the machine are two of the most important ones.

4.6.1 Capacity cone

To simplify our expression, here we take a head pivot-able and table rotational machine as an example, assume the rotation axis of the table is parallel to Z-axis. As mentioned in Chapter 4, all geometric elements of a SDV are represented in the stock reference system xyz. The z-axis of xyz is initially parallel to the Z-axis of the machine system. In the following part of this section without specified all analyses are based on xyz system.

As we know that angular motion about pivot axis results an angle between the tool axis and z-axis. By system transformation we can change any rotation about the pivot axis to revolving about a line parallel to the pivot axis and passing the tool tip, this line is called the shift pivoting axis. We can image that the tool tip is fixed and the tool can turn around the tip around the shift pivoting axis. Ignore the diameter of the tool, pivot motion of a tool sweeps a circle sector in a vertical plane. The circle sector bounded by unit vectors representing the two extreme pivoting positions of the tool axis is defined as the pivoting capability sector of a machine, denoted PCS, which is symmetric about the vertical line passing the tool tip. Rotating half of PCS 360° about the vertical line, we get a cone (or a section of cone if the table can only rotates less than 360°) volume within which all points are accessible by a tool without any linear motion. This cone is named the pivoting capability cone of a machine. The vertex angle of a capability cone equals to double value of the maximum pivot angle (the pivoting capability of a machine) which is denoted

Amax_pivot. The axis of the cone is parallel to Z-axis of the machine, see Figure 4-11. For better visual effect the cone was made in such a way that Amax_pivot is an acute angle.

At any given point p(a, b, c) in face f of a SDV, whose position vector is denoted **p**, a capacity cone can be created. Denote *x* a point in the capacity cone and **X** is its position vector, vector $\mathbf{X}-\mathbf{p}$ must locates within the capacity cone associated with point p. Denote **d** the unit vector of $\mathbf{X}-\mathbf{p}$, that is to say that the capacity cone is consisted of points ***x*** such that the dot product of $\mathbf{d}=\mathbf{X}-\mathbf{p}$ and $\mathbf{k}$ (unit vector of z-axis) is positive and not less than $\cos(\text{Amax\_pivot})$, that is

$$\text{cone C: } 0 \le \cos(\text{Amax\_pivot}) \le \mathbf{k}\frac{\mathbf{X}-\mathbf{p}}{\|\mathbf{X}-\mathbf{p}\|},\ f(\mathbf{p})=0, \mathbf{X}\in\mathbf{B}, \mathbf{X}\neq\mathbf{p} \qquad (4.63)$$

where **B** is the set of position vectors of points in the hemisphere enclosing the whole volume of the final stock.

To machine a face of no more than one-dimension curvature, both the tip and the side cutting edge of a tool can be used. While to produce a face of two-dimension curvature the tip of a tool with a hemispherical end (also called ball nose mill) is necessary.

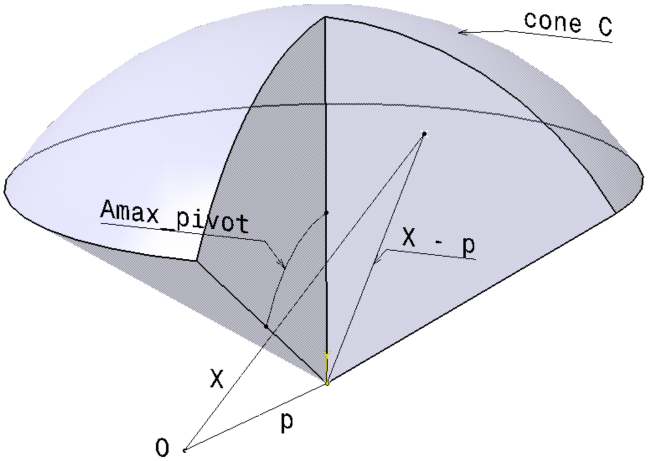


Figure 4-11 Pivoting capacity cone

4.6.2 Tool posturing plane

Given a point p(a, b, c) in face f of a SDV, the normal vector of face f at p pointing outwards the volume is denoted **n**. Let VL denotes “vertical line” which is the line passing p and parallel to Z-axis of the machine. We define the angle between **n** and the vertical line the deflection angle θ of f at the given point p. We also define the tool axis vector ***s*** which is a unit vector originated at the center of tool tip, coincident with tool axis and directing inward the tool. No matter which cutting edge of an endmill is used, satisfaction of the following two conditions is necessary for a part face of a SDV to be completely machinable in one setup.

1) At any point in the face the angle between its normal vector **n** and tool axis vector **s** must be no less than $\pi/2$. The smaller one of the two angles formed by the two vectors is defined as the angle between the two vectors, thus it is no greater than 180°.

2) For any portion of the face there is a direction from which the face is accessible by a tool which is reasonably slim and long enough.

Now we discuss the mathematical expression for the first requirement. As having been mentioned in Chapter 3, in advanced B-rep representation, each face is a subset of a surface defined by a geometric representation. In other words a face f is given implicitly as a set of points (x, y, z) satisfying f(x, y, z) = 0, $(x, y) \in D$. At point p(x, y, z) in face f(x, y, z) the unit normal vector **n** is determined as the gradient of the function f(x, y, z)

$$\mathbf{n} = \frac{\nabla f(x,y,z)}{\left|\nabla f(x,y,z)\right|} = n_x\mathbf{i} + n_y\,\mathbf{j} + n_z\mathbf{k} \qquad (4.64)$$

Where **i**, **j**, **k** are standard unit vectors of xyz system. The positive direction of a SDV face obeys the right-hand rule when the fingers of the right hand are curled to match the counter-clockwise path of the face loop.

Given face f(x, y, z) = 0 of a SDV, its unit normal vector at point p(a, b, c) is denoted as $\mathbf{n} = [n_x\ n_y\ n_z]^T$. The z component n_z of **n** equals to the value of Cosine of the angle θ that vector **n** deflects from z-axis, $\theta = \arccos n_z$. A positive n_z means that the face at given point is facing in the direction of positive z-axis, otherwise it faces toward the negative z-axis. A face having negative n_z has to be accessed by a tool above the face, which means the tool has to situate in the positive half space of the plane orthogonal to vector **n**, while positive n_z implies that the point is approachable from a lower position, thus a tool has to locate in the negative half space of the plane orthogonal to vector **n**.

Not lose generality, suppose that z-axis is parallel to Z-axis when a workpiece is initially positioned, thus θ is also the angle between **n** and Z-axis. In the case that the initial z-axis is not parallel to Z-axis, coordinate transformation can be made so that the geometries of the SDV are defined in a system of which the z'-axis is parallel to Z-axis. To save space the issue of coordinate system transformation will not be discussed in here.

If the side cutting edge of an endmill is used to machine the neighborhood of point p in face f, f ∈ SDV.face, tool axis vector **s** should be orthogonal to the normal vector **n** of face f at p. Thus **s** must situate in the plane perpendicular to **n**. The plane P passing p and orthogonal to vector **n,** named tool posturing plane, can be represented by

$$P:\ \mathbf{n}\cdot(\mathbf{X}-\mathbf{p})=0, \mathbf{X}\in\mathbf{B}, \mathbf{X}\neq\mathbf{p} \tag{4.65}$$

Where X is the position vector of an arbitrary point $\mathscr{X}$(x, y, z) in plane *P*, **p** is the position vector of point p, see Figure 4-12.

If the tool tip is used to machine face f at p, then the reverse of **n** should within the capacity cone defined in criterion(4.63).

4.6.3 Tool posture flat/space (TPF/TPS)

The intersection of plane P and cone C determines the sector within cone. This sector reflects the pivoting capacity available for machining the neighborhood of p with the side cutting edge of a tool.

If we translate the intersection by placing the cone vertex at one of the access points p_0 of the SDV, we get the area available for positioning a tool initially to machine the SDV. And the target position p can be reached theoretically by the tool with translation motion only. Replace p in equations (4.63) and (4.65) with the geometrical center p_0 of the corresponding open face of the SDV, we obtain plane P0 perpendicular to **n** and passing p_0 as well as the capability cone C0 having p_0 as its vertex, that is

$$P0:\ \mathbf{n}(\mathbf{X}-\mathbf{p}_0)=0,\ \mathbf{X}\in\mathbf{B},\ \mathbf{X}\neq\mathbf{p}_0 \tag{4.66}$$

$$C0:\ \mathbf{k}\frac{\mathbf{X}-\mathbf{p}_0}{\|\mathbf{X}-\mathbf{p}_0\|}\leq\cos(\text{Amax\_pivot}),\ \mathbf{X}\in\mathbf{B},\ \mathbf{X}\neq\mathbf{p}_0 \tag{4.67}$$

The intersection of $P0$ and $C0$ is the planar area in which we can initially position a tool with its tip at point p_0. Name this common area as TPF (Tool Posture Flat), then we have

$$\text{TPF} = P0 \bigcap C0 \qquad (4.68)$$

TPF is the flat area that a machine can provide for positioning a tool to cut the workpiece with its side edge at point p. In other words tool axis vector **s** must situate in this sector and parallel to the intersection of plane P and face f, see also Figure 4-12.

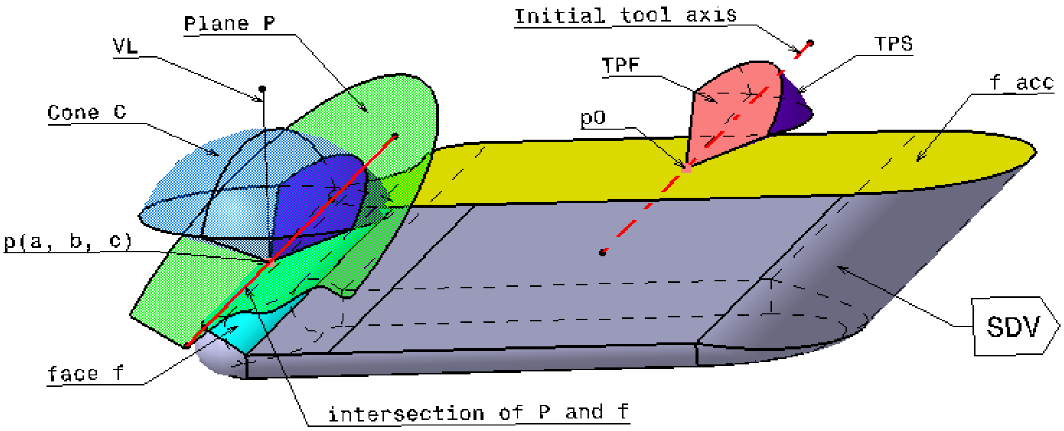


Figure 4-12 Tool posture flat/space

If not the side edge but the tip of a tool will be used to machine the neighborhood of p, the tool is required to be situated in the negative half space H of the tangent plane passing p, if $n_z \geq 0$, or in the positive half space if $n_z \leq 0$. As situating in the positive half space requires smaller pivoting angle than in the negative one, we consider only the tool in the negative half space. The negative half space is determined by

$$H: \ \mathbf{n}(\mathbf{X}-\mathbf{p}) \leq 0, \ \mathbf{X} \in \mathbf{B}, \ \mathbf{X} \neq \mathbf{p} \qquad (4.69)$$

The capability cone C remains the same as represented by equation (4.63). Similarly to equations (4.66) we have the negative half space determined at point p_0

$$H0: \ \mathbf{n}(\mathbf{X}-\mathbf{p}_0) \leq 0, \ \mathbf{X} \in \mathbf{B}, \ \ \mathbf{X} \neq \mathbf{p}_0 \qquad (4.70)$$

The capacity cone will be no change as defined in inequality (4.67)

The common part of $H0$ and $C0$ determines the subspace TPS (Tool Posture Space) for positioning the tool

$$\mathrm{TPS} = H0 \bigcap C0 \tag{4.71}$$

For better view, in Figure 4-12 cone C was defined in a domain smaller than **B**.

TPS represents the space area that a machine can provide for placing a tool initially to cut the neighbor area of point p in face f of a SDV with the tool tip. It is intuitive that TPF is a subset of TPS, and from the perspective of geometry TPF is one of the boundary faces of TPS.

4.6.4 Pivoting control point

Now we are going to find the point in face f where the required pivot angle is the greatest as compared to those at other places in f. This point is named as the pivoting control point. In later sections of this research “Deflecting control point” is used as an equivalent to pivoting control point.

Before further discussing we introduce two new concepts: sheltering and sheltered faces which are the places where the greatest pivot angle may exist.

Given two faces f and g of a SDV, in face f there is a point where the normal vector of f has z component no less than zero, and face g there is a point where the normal has negative z component. The projections of face f and g onto xy-plane are overlapped, partly or completely, and the area of the common part of these projections is positive, , if one of the vertices of face f is higher than all vertices of face g, then face f is the sheltering face of g, and face g in turn is the sheltered face. That is given $\mathrm{f, g} \in \mathrm{SDV\_face},$ if the following conditions are satisfied,

1) $\left\{\mathrm{n}_z(p) > 0, p \in \mathrm{f}\right\} \neq \left\{0\right\}$, and $\left\{\mathrm{n}_z(q) < 0, q \in \mathrm{g}\right\} \neq \left\{0\right\}$,
2) $\mathrm{area}\left\|\mathrm{proj}_{xy}(\mathrm{f}) \bigcap \mathrm{proj}_{xy}(\mathrm{g})\right\| > 0$,
3) $\max(z_{f.vert}) \geq \max(z_{g.vert})$

then face f is the sheltering to face g, and face g is the sheltered face.

When a face is sheltered by a part face, both sheltering and sheltered faces are not accessible from the downward direction parallel to z-axis. In this situation it is necessary to pivot the tool or tilt the workpiece so that the faces are accessible by a tool. The next question to be answered is how many degrees the tool is required to pivot. As equivalent relative position of a tool and a SDV can be get as a result by either pivoting a tool or tilting the workpiece give, for description convenience we take head pivot-able machine with rotatable table as an example in our discuss.

A perfect sheltering face has normals of positive z-components everywhere in the face, while in a perfect sheltered face z-component of normal vector at any point is negative. A face having both positive and negative normal is a fold face which will be discussed later in section 4.5.6.

To find the pivoting control point, we are interested only in part faces. To machine a perfect sheltering face, a part face of a SDV, side edge cutting requires a tool located in the plane perpendicular to **n**, which means the smaller angle that **n** deflects from z-axis, the greater pivoting angle of a tool is required. In tip cutting mode a tool must located in the negative half space of the plane perpendicular to **n**. Thus at the same point in sheltering face, tip cutting requires a tool to pivot an angle no less than that required by side edge cutting mode. In Figure 4-13 faces f1 and f2 are perfect sheltering faces to be machined with the side cutting edge of a tool, As n1 (the normal of f1 at given point) has greater deflection angle than n2, so f1 requires a tool to be postured with a less pivoting angle than f2 does. Thus tool axis vector s1 deflects smaller angle from the z-axis than s2 does. For better visual effect, instead of in the same face, points in different faces are taken as examples in Figure 4-13.

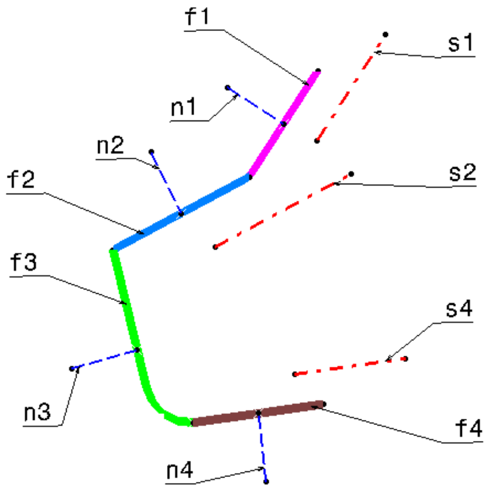


Figure 4-13 Tool postures for side edge cutting mode

On the contrary any place in a sheltered face requires less pivoting angle when it is machined by the tip than by the side edge of a tool. Thus the maximum reasonable pivot angle may exist at the point requiring side edge cutting and where the normal vector deflects the greatest angle from z-axis (the smallest $n_z \leq 0$), or the place where **n** deflecting a smallest angle but it requires the tool axis be most nearly coincident with the **n** (at the meantime $\mathbf{s} \bullet \mathbf{n} \leq 0$) as shown in Figure 4-14.

As a conclusion the following three kinds of points are taken as pivoting control points. 1) For a sheltering face the point where the face normal has minimum deflecting angle. 2) For a sheltered face the point where the normal has the smallest deflecting angle, and 3) Also for a sheltered face the point where the normal has the greatest deflecting angles. Evaluation of angular accessibility of related faces will be discussed based on these pivoting control points.

The z component of unit normal vector **n** represents the Cosine value of the angle that **n** deflects from positive z-axis. For any face the minimum deflection angle occurs at the point where the normal vector has maximum z component. That is

$$\theta_{\min} = \arccos(\max(n_z)) \qquad (4.72)$$

For a sheltered face the maximum deflection angle happens when n_z reaches its minimum value. That is

$$\theta_{\max} = \arccos(\min(n_z)),\ n_z \leq 0 \qquad (4.73)$$

By determining the $\max(n_z)$ and $\min(n_z)$ for sheltering and sheltered part face respectively we can find the pivoting control points.

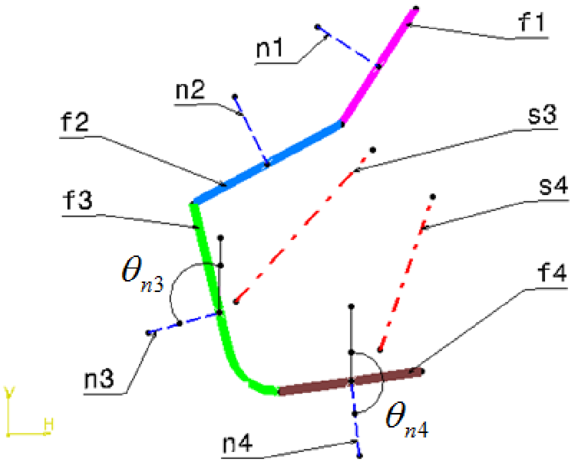


Figure 4-14 Tool postures in tip cutting mond

If in a part face of a SDV the z component of normal vector changes from positive to negative, and vice versa, as a point moves downward along a curve in the face, the face is self-sheltered, the higher portion of the face shelters its lower portion, and they are named sheltering portion and sheltered portion respectively. For a self-sheltered face both deflection control points are located in the same face. In sheltering portion the pivot control point p1 has a normal vector of maximum z component, while at the control point p2 in the sheltered portion the normal vector has minimum z component.

4.6.5 Required pivot angle

Given a part face f_i of a SDV, at any point in f_i the z-component of the unit normal vector can be defined as

$$n_z(x,y,z) = \frac{1}{|\nabla f_i|} \cdot \frac{\partial f_i}{\partial \mathrm{z}}, (x,y,z) \Big| f_i(x,y,z) = 0 \qquad (4.74)$$

Mathematical principle to determine the maximum and minimum values of n_z will not be repeated here. Suppose p_1 and p_2 are two points in f_i and

$$\begin{aligned} n_z(p_1) &= \max(n_z) \\ n_z(p_2) &= \min(n_z) \end{aligned} \qquad (4.75)$$

From the results of equation (4.75) face f_i can be identified as

a) sheltering face, if $n_z(p_1),\ n_z(p_2) \geq 0$;

b) sheltered face, if $n_z(p_1),\ n_z(p_2) < 0$;

c) self-sheltered face, if $n_z(p_1)n_z(p_2) < 0$; and

d) flat or vertical rotation face with linear, if $n_z(p_1) = n_z(p_2)$.

In situation a) the TPF/TPS, as discussed in section 4.6.3, will be determined in accordance with $\mathbf{n}(p_1)$, in situation b) in accordance to $\mathbf{n}(p_2)$. In situation c) the TPF/TPS should be determined in accordance to both of these points. In situation d) the TPF/TPS can be determined according to either one of the two points.

At any point of p_1 and p_2, if $\{\text{TPF/TPS}\} - \{p_1, p_2\} = \{0\}$, which means the required pivoting angle is out of the capacity cone, then the machine cannot provide a tool with the direction required by the face. In this case we say that the SDV is not operable angularly on the machine. Any SDV corresponding to a part is found angularly inoperable, then the part is not machinable by the machine.

Suppose SDV is angular operable, thus

$$\left[\{\text{TPF/TPS}\} - \{p_1, p_2\}\right]_{f_i} \neq \{0\},\ f_i \in \text{SDV.f.part} \tag{4.76}$$

where SDV.f.part is the set of part faces of the SDV. As we know TPF/TPS corresponding to face *f* determines a subspace of vectors originated at SDV.p_acc, without consider blocking by other part faces, face *f* is accessible by a tool oriented along any vector in the TPF/TPS. The heads of the vectors in TPF/TPS are in the spherical face, and their tails are coincident with the center of the sphere, which is the access point. Any nonempty intersection of all TPF/TPS of a SDV determines the vector subspace, denoted CTP, all part faces of the SDV is accessible by a tool oriented parallel to any element of CTP. That is

$$\text{CTP} = \bigcap \{\text{TPF/TPS}\}_{f_i},\ f_i \in \text{SDV.f.part} \tag{4.77}$$

If $\text{CTP} \neq \{0\}$, all part faces of the given SDV is accessible from a common direction. In CTP the vector of smallest deflection angle (from +z-axis) can be determined by making a plane PS vertical to xy-plane and passing the center vertex p_0 of CTP, the intersection of PS and CTP has two boundary edges having the same vertex p_0. The one that has smaller deflection angle gives the direction of a tool, and this angle is defined as the required pivoting angle.

If $\text{CTP} = \{0\}$, at least one part face of the SDV has to be accessed from a direction different from that of other faces. In this case a direction can be determined so that from the direction as much as possible volume of the SDV can be removed in one operation. The SDV can also be machined from several directions determined by $\{\text{TPF/TPS}\}_{f_i}$. Detail discussion on this topic is left for future research.

4.7 Linear accessibility and collision

In section 4.6 we have discussed the angular operability of a SDV, in this section, we are going to develop a method to identify the linear accessibility of a part face in a SDV. In this work we define the linear accessibility as that from a given direction a tool, reasonably slim and long enough, can reach every point in the face without collision with any other face of the part to be machined.

Given two part faces *f* and *g*, suppose $f \in \mathrm{SDV.f.part}$, face *g* can be the boundary of the same SDV or not. If a tool will collide face *g* in the process to approach face *f* in the direction $-\mathbf{s}$, then while viewed in direction $-\mathbf{s}$, both faces must be overlapped totally or partially, and the overlapped portion of face *g* must be located closer to the viewer than that of face f. If no part face will be impacted from direction $-\mathbf{s}$, then face f is linearly accessible from $-\mathbf{s}$.

Based the intuitive analysis above, make plane Q perpendicular to **s** and passing the farthest vertex q of the final stock viewed in direction **s**. Denote f_{Qs}, g_{Qs} the projections face *f* and *g* onto plane Q in direction **s** respectively,

$$\begin{aligned} f_{Qs} &= project(f,\ \mathrm{Q}, \mathbf{s}) \\ g_{Qs} &= project(g,\ \mathrm{Q}, \mathbf{s}) \end{aligned} \tag{4.78}$$

denote $\mathrm{inter}f_{Qs}g_{Qs}$ the intersection of f_{Qs} and g_{Qs} in plane Q**,**

$$\mathrm{inter}f_{Qs}g_{Qs} = f_{Qs} \cap g_{Qs} \tag{4.79}$$

If $\mathrm{inter}f_{Qs}g_{Qs} = \{0\}$, face *f* is not blocked by *g* from direction –**s,** thus there is no collision with ***g*.** If $\mathrm{inter}f_{Qs}g_{Qs} \neq 0$, let A_{fQ}, A_{gQ} be the portions of faces *f* and *g* whose projections onto Q are overlapped, that is and the area of the intersection $\left\|\mathrm{inter}f_{Qs}g_{Qs}\right\|_2 = 0$, then there is no collision. If $\left\|\mathrm{inter}f_{Qs}g_{Qs}\right\|_2 \neq 0$, let A_{fQ}, A_{gQ} be the portions of faces *f* and *g* whose projections onto Q are overlapped, that is

$$project(A_{fQ}, \mathrm{Q}, \mathbf{s}) = project(A_{gQ}, \mathrm{Q}, \mathbf{s}) = \mathrm{inter}f_{Qs}g_{Qs}$$

let D_{fQ}, D_{gQ} denote the distance between plane Q and points in A_{fQ} and A_{gQ} the farthest away from Q respectively. Denote d_{fQ} , d_{gQ} the smallest distances between plane Q and A_{fQ} , A_{gQ} respectively. Thus we get the criteria for no collision

$$\begin{aligned}
&\text{inter} f_{Qs} g_{Qs} = \{0\} \\
&\text{or} \\
&d_{fQ} < d_{gQ}, \text{ if inter} f_{Qs} g_{Qs} \neq 0 \qquad (4.80) \\
&\textit{or} \\
&D_{fQ} \leq D_{gQ}, \text{ if } d_{fQ} = d_{gQ} \text{ and inter} f_{Qs} g_{Qs} \neq 0
\end{aligned}$$

If inter$f_{Qs} g_{Qs} \neq 0$ and $d_{fQ} > d_{gQ}$ or even worse $D_{fQ} > d_{gQ}$, collision will happen when the tool approaches face *f* in direction –**s**. To avoid collision other directions in the TPF/TPS corresponding to *f*, or other accessing points of the SDV should be tried. If collisions exist in all attempts then the face is not machinable with given machine.

4.8 Assign machining operation to a SDV

When the access face and corresponding tool axis vector is determined, a SDV can be analyzed for machining operation assignment.

Let $\{f_i\} =$ SDV.f.part, $i = 1,3,...,N$ the set of part faces of a SDV, denote **s** the access direction, $\{p_i\}$ the set of geometric centers of $\{f_i\}$ and $\{\mathbf{n}_i\}$ the set of normal vectors of faces $\{f_i\}$ at $\{p_i\}$ respectively.

First of all, if $N = 1$, the SDV has only one , a contouring operation is assigned.

If $\exists f_j \in \{f_i\}$, that the tangent plane *Q* of f_j at p_j separates the boundary vertices of the SDV into two groups. Without loss of generality, suppose the non-empty set of vertices in the positive half space of *Q* is consisted of non-part vertices, thus all part faces are in the negative half space of *Q*.

If faces in $\{f_i\}$ are adjacent one the other, and every $f \in \{f_i\}$ has at least two non-adjacent edges shared by two non-adjacent open faces respectively, especially if

$$\prod \mathbf{n}_i > 0$$

then the SDV can be machined by tool sweeping these faces, thus a contouring operation can be assigned to the SDV. If $\prod \mathbf{n}_i = 0$, but there is at least two non-part faces in the boundary of the SDV that adjacent to all faces in $\{f_i\}$, then the SDV can also be machined by a contour operation.

If $N = 3$, and at least two part faces, $f_1, f_2 \in \{f_i\}$ for example, are planar and have opposite normal directions, that is

$$\mathbf{n}_1 \cdot \mathbf{n}_2 = -1$$

Then this SDV is a slot. In this work, the two opposite planar faces are not required to be parallel.

What we defined here is only a straight slot. In fact some slots can be curved. Developing approaches to identifying curved slot is left for future study.

It is noticed that through hole and through pocket have two opposite open faces enclosed by part edges. Denote $\{g_i\}, i \le M$ the set of non-part face of SDV, and $\{\partial g_i\}, i \le M$ the set of enclosures of $\{g_i\}$. If $\exists g_j, g_k \in \{g_i\}$, such that $\mathbf{n}_j \mathbf{n}_k < 0$ and

$$\partial g_i = \{g_i\} \cap \{f_i\}, i = j, k$$

Then through hole or through pocket operation can be assigned to the SDV.

If $M = 1$, the SDV is a blink hole or pocket.

If $M \ge 2$, the SDV, every open face of a bottom closed pocket SDV is adjacent to at least two non-adjacent part faces. In the contrary every of a step SDV is adjacent to at least two non-adjacent open faces. Denote $\{g_i\}, i \le M$ the set of non-part face of SDV. If $N \ge 2$ and $M \ge 2$, and if $\exists g_j, g_k \in \{g_i\}$, $f \in \{f_i\}$ such that $g_j \cap g_k = \varnothing$, when $g_i \cap f \ne \varnothing, i = j, k$ and $\mathbf{n}_j \mathbf{n}_k < 0$, then a step operation can be assigned to the SDV. Otherwise, pocket operation. Another criteria for distinguish a pocket from a step can be stated as follows. Denote J_p and J_{op} the numbers of adjacent part faces and adjacent non-part faces of SDV respectively. It is most often that when $J_p > J_{op}$, the SDV is a pocket, otherwise it is a step.

Operations discussed above, together with facing, are most often used in roughing an aeronautical structural part. Associating other operations with SDVs is left for future study.

4.9 Tool length, Diameter and Tip

When all part faces are linear accessible from corresponding directions, our next task is to determine the parameters (length, diameter, tip radius) of tools suitable for roughing or face machining. In following discussion main factors, which constrain the parameters of a tool, such as 1) the geometry of a face to be machined, 2) the direction of tool axis **s**, 3) the position of the face in respect to other faces of the part, and 4) the volume of the final stock are taken into account.

4.9.1 Tool length

Each SDV may require different tool length. As every point in a SDV has to be reached by a tool, thus the tool length is determined by the point *p*, $p \in S\mathrm{DV}.f$, which is the point farthest away from corresponding access point in direction **s**. To determine the tool length, draw plane *P* passing *p* and perpendicular to **s**, draw plane *Q* passing *p*, and $\mathbf{n}_Q = \mathbf{s} \times \mathbf{k}$, **k** is the standard unit vector of z. The positive direction of *P* is **s.** Denote **X** the set of points which are located on the intersection of *Q* and boundary faces of the final stock SF, and in the positive half space of *P*, that is

$$\mathbf{X} = \left\{ \mathbf{x} \middle| (\mathbf{x} - \mathbf{p})\mathbf{s} > \mathbf{0},\ \mathbf{x} \in Q \bigcap \mathrm{SF.f} \right\} \tag{4.81}$$

where SF.f is the set of boundary faces of the final stock. Denote L the line passing *p* and parallel to **s,** *q* the intersection of **X** and L

$$\mathbf{q} = \left\{ \mathbf{x} \middle| \frac{(\mathbf{x} - \mathbf{p})\mathbf{s}}{\|\mathbf{x} - \mathbf{p}\|} = 1,\ \mathbf{x} \in \mathbf{X} \right\} \tag{4.82}$$

where **p, q** are the position vector of point *p* and *q* respectively. Then the distance *d* between *p* and *q* is the minimum required length of the tool.

$$d = \|\mathbf{q} - \mathbf{p}\| \tag{4.83}$$

The length of the intersection of L and the SDV is the minimum required length of cutting edge.

4.9.2 Tool diameter

For roughing machining it is preferred that a tool has a diameter as big as possible, while for face machining, many conditions constrain the tool diameter, in this work only geometry factors are taken into account.

To roughing a bottom narrowed SDV, a straight column SDV and hole SDVs, the possibly big diameter can be determined as follows.

Plane Q perpendicular to **s** and passing the access point p_{acc}, denote E_B the set of edges of the bottom face, E_Q the projection of E_B onto Q in direction **s**. Let D_Q the set of distances between p_{acc} and elements of E_Q. Thus the required diameter D must be no greater than two times of the minimum of E_Q

$$\frac{D}{2} \le \min(\mathrm{E}_Q) \tag{4.84}$$

This equation is applicable to both bottom narrowed SDV and straight column SDV. For a hole SDV, the diameter of the hole should be the upper limit of tool diameter.

For a general formed SDV, we need the function to find the distances d between the part faces of the SDV and the line L coincident with **s**. The tool diameter must be inferior to two times of the smallest distance.

$$\frac{D}{2} \le \min(d \,|\, d \in dd) \tag{4.85}$$

where $dd = \{\text{surface\_curve\_solver(SDV.f.part, L}\}$ is the set of distances between L and part faces SDV.f.part. Mathematical principles for finding the curve/surface distance are not repeated here.

If another sub delta volume SDV[j] whose face g intersects with L is point A

$$\{A\} = \mathrm{L} \cap \mathrm{SDV[j].g} \neq \{0\} \tag{4.86}$$

and point A is closer to point q, defined by (4.82), than point p_{acc} does, then tool diameter determined in accordance to SDV[j] must be the upper limit of D defined above in conditions (4.84) and (4.85). Detail discuss is omitted.

To save space, tool diameter for SDV that requires a tool of special form, such as a T slot, will not discussed in this work

4.9.3 Type of tool tip

The tool tip is constrained not only by the geometry of a surface to be machined. It is the combination effect of surface geometry and tool direction that determines the tool tip to be used.

Given point *p* in face *f* of a SDV, denote **n**(*p*) the normal vector of face *f* at *p*, **s** the tool axis vector. Point p can be the geometric center of the face.

If $\mathbf{ns} = -1$, the tool is perpendicular to *f* at *p*, and if face *f* is convex, a ball end mill has to be used, the tip radius of tool can be at most equal to the smaller radius of curvature corresponding to the principle curvature k_1. A concave face does not limit the tip radius. If face *f* is planar, the tool tip can be flat.

If $\mathbf{ns} = 0$, the tool is tangent to face *f* at *p*, the tool tip can be in any shape (ball end, flat end or rounded corner). Furthermore if face *f* is concave and $\mathbf{n} \times \mathbf{s}$ determines one of the planes of principle curvature, the tool has to be either a ball ended or a round corner one.

If $-1 < \mathbf{ns} < 0$, the tool is neither perpendicular nor parallel to *f* at *p*, in this case, if face *f* is planar, a flat end mill with round edge can be used. If face *f* has two dimensional curvatures, a ball end mill or is required.

The diameter of a ball end mill is limited by the radius of curvature of the face to be machined. Theoretically the minimum radius of curvature of *f* can be determined by maximizing the principle curvature k_1, but the calculation maybe time-consuming. The curvature formulas for implicit curves and surfaces can be found in [99]. A simpler way suggested here to find approximately the maximum curvature of face *f* is as follows.

Define curves $\{c\}$ the intersections of planes $\{P\}$ passing the middle points $\{p_c\}$ of the part edges $\{e\}$ of f and perpendicular to the individual edges respectively. Each c is represented implicitly by the surface function $F(x, y, z) = 0$ with the condition $(x, y, z) \in f \cap P$. The curvature k_c of c can be deduced from $F(x, y, z) = 0$, then find the maximum $k_{c\max}$ of k_c which is related to the radius of curvature r_c by $k_{c\max} = 1/r_c$. The minimum element of $\{r_c\}$ can be taken as the upper limit of the tip radius. Precise method for finding the minimum radius of curvature of a surface is left for future research.

If face *g* shares a common edge with face *f* which has two dimension curvatures, then in the area neighbouring to the common edge, face *g* must be also machined with the ball end mill determined by face *f*.

For the purpose to machine a SDV with one operation, the priority order for selecting a tool is ball end, round edge and flat end. But for machining efficiency, it may be better to use tools of different tips for individual part faces respectively. This topic is left for future research.

4.10 Operation sequence generation

By procedures of delta volume decomposition and machining attributes analysis, the geometry and essential information of machining were associated within individual elements of a complete set of SDVs. The next step is to put the so obtained SDVs in an ordered list so that cutting operations can be conducted one after another. One of the objectives of this research is to develop approaches to sequencing the SDVs in a reasonable order so that corresponding operations can be performed in the same order. The resulted list of operations is not optimized, but it can be reordered by a process programmer according to his experience. Machining attributes of related SDVs that may be impact significantly by reordering are: access points and lengths of cutting edges. The optimization and impact analysis are left for future study.

To sequence operations for roughing the individual SDVs can be organized in the following ways.

1. Facing first

   If there is a facing SDV, it will be the first in the list

2. Priority of interior

All interior SDVs are selected to be list before other exterior SDVs except the facing SDV

3. Priority of accessible from Z

   Denote G$_{\text{pos}}$ a subset of SDVs, an elements in G$_{\text{pos}}$ has access vector parallel to unit vector **k.** Denote **V**$_{\text{acc}}$[i] the set of access vectors of SDV[i], if $\exists \mathbf{s}_j \in \mathbf{V}_{\text{acc}}[\text{i}]$ that $\mathbf{s}_j\mathbf{k} = 1$, then this SDV[i] is put into G$_{\text{pos}}$, otherwise it is grouped in G$_{\text{piv}}$. The subscript pos means along "positive z" and piv signifies "pivot to z".

   For roughing purpose only, looser constrain can be used to replace the criteria $\mathbf{s}_j\mathbf{k} = 1$. For example if $0 < \mathbf{s}_j\mathbf{k} < 1$ and the most portion of the volume is accessible from **k**, the SDV[i] can also be grouped in G$_{\text{pos}}$. Similar to Algorithm 4.2, replace vector **n** by **k,** if

   $$\frac{\left\|\text{proj}(\bigcup \text{SDV[i].f, Q, } \mathbf{k}) - \text{proj}(f_{acc}, \text{Q, } \mathbf{k})\right\|}{\left\|\text{proj}(f_{acc}, \text{Q, } \mathbf{k})\right\|} < \delta < 1 \qquad (4.87)$$

   Then the SDV[i] is to some extents machinable from Z. The smaller the value of δ, the more volume can be removed from Z. Thus every SDV satisfying condition (4.87) can also be put in G$_{\text{pos}}$.

4. Priority of higher position

   Arrange SDVs in Gpos in the order that higher position first. The SDV that has highest vertex will be the first in the list.

5. Priority of larger access face

   If both SDV[j] and SDV[m] are measured the same height, the one has larger access face will be placed preceding the other in the list, so that the workpiece is more rigid when removing the SDV of bigger cutting area.

6. Priority of closer to the original point/reference plane

If two SDVs have the same height and same area of access face, the one that has a vertex closer to the original point or a reference plane will take a place before the other in the list.

7. Priority of inclined hole

   SDV corresponding to an inclined hole should be put after the SDV sharing its access face, but before all other SDVs adjacent to it.

8. SDVs in G_{piv} will be sequenced according to the following precedence.

   a. +X → +Y → -X → -Y

      A SDV whose access vector is in accordance with unit vector **i** and whose access point is closer to the plane perpendicular the X axis and passing vertex *p* on the diagonal of the stock, $p_x > 0$ takes the position before others; after sequencing all SDV accessible from +X, sequence those accessible from +Y and so on.

   b. Priority of height

      SDVs accessible from the same direction and has the same distance to corresponding plane are listed according to their height. Higher one will be listed first. Hereafter, priorities 6 and 7 mentioned above are applicable.

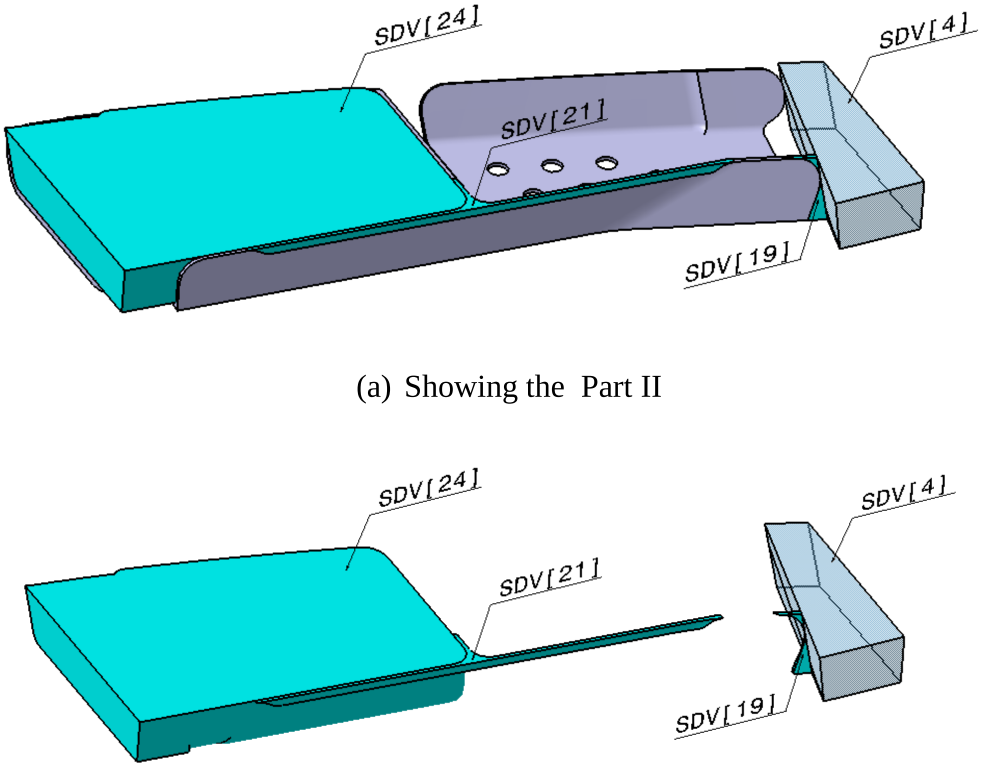
A complete operation sequence is the list of operations in the same order as their corresponding SDVs are listed in G_{pos} followed by those in G_{piv}.

Modification of the sequence may require longer cutting edge of a tool if a lower SDV is to be machined before a higher one. Thus the relative positions in the list of two SDV, one adjacent with the other at its access face, should be verified while determine the tool parameters.

CHAPTER 5 APPLICATION OF ATTRIBUTES ANALYSIS

To verify the applicability of the attributes analysis approaches developed in the previous sections of this work, we take some of the SDVs of part II as examples shown in Figure 5-1. These SDVs are obtained by delta volume decomposition procedures as described in chapter 3. To see the relative positions the part is shown up in Figure 5-1 (a). The part was hidden in Figure 5-1 (b) so that the small SDVs can be seen more clearly.

As shown in Figure 5-1, SDV[4] is an exterior SDV, and the others SDV[19], SDV[21] and SDV[24] are interior SDVs. We will explain how our approaches are applied to determine the machining attributes of these SDVs, which are the access face, access point, access direction, and required pivot angle. Tool parameters (diameters, type of tool tip, and length) will be determined according to geometry and machining attributes of each corresponding SDV, and also according to operation sequence.



(a) Showing the Part II

(b) Without showing the Part II

Figure 5-1 Relative positions of example SDVs of Part II

5.1 Access face

In this section we will apply the approaches developed in section 4.2 to SDV[4] first to determine its set of access faces. The six faces of SDV[4] are defined in Figure 5-2.

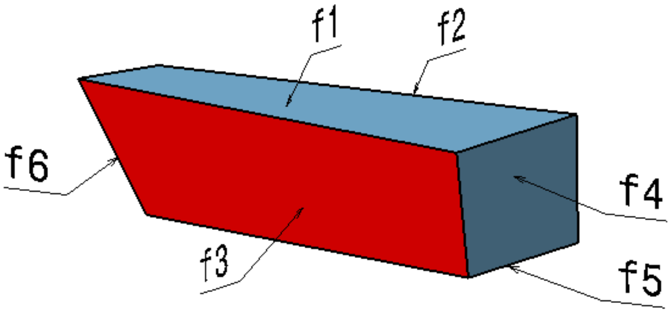


Figure 5-2 Boundary faces f_1, f_2,…, f_6 of SDV[4]

Denote F_4 the set of boundary faces of SDV[4], that is

$$F_4 = \{ f_i \},\ i = 1, 2, ..., 6 \tag{5.1}$$

As it is shown in Figure 5-3 face f_3 is the only boundary face of SDV[4] that intersects the Part II. the intersections are two points q_1 and q_2 in its boundary face f_3. That is as

$$F_4 \cap P_B = f_3 \cap P_B = \{q_1, q_2\} \neq \varnothing \tag{5.2}$$

where P_B denotes the solid Part II, f_3 is the close face of SDV[4]. To determine the open faces of SDV[4] apply equation (4.1) to SDV[4] we get

$$\begin{aligned} \text{SDV[4].open\_face} &= F_4 -^{*} \bigcup f_j \,\middle|\, f_j \in F_4 \wedge f_j \cap P_{\mathrm{B}} \neq \varnothing,\ j = 1, 2, ..., 6 \\ &= F_4 - f_5 = \{ f_1, f_2, f_4, f_5, f_6 \} \end{aligned} \tag{5.3}$$

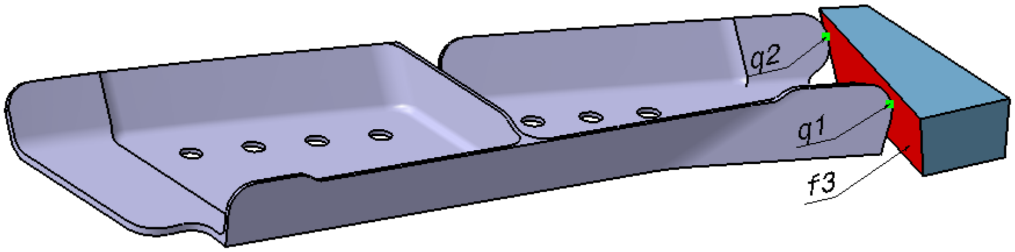


Figure 5-3 Intersections of SDV[4] and Part II

And theoretically, according to equation (4.2), any element face of the set SDV[4].open_face can be chosen as the access face of the SDV. That is

$$\text{SDV[4].f\_acc} = \text{SDV[4].open\_face} = \{ f_1, f_2, f_4, f_5, f_6 \} \tag{5.4}$$

In equation (5.3) P_{B} instead of P_{B}.face is used because no SDV can intersect the interior of a part, thus the Boolean operations $f_j \cap P_{\mathrm{B}}$ and $f_j \cap P_{\mathrm{B}}.\text{face}$ will give the same results.

Another example is given by showing the open faces of SDV[19].

As shown in Figure 5-4 the boundary of SDV[19] consists of ten faces f_1 to f_{10}. Among them are faces f_1, f_2, … , f_5 which have intersections with the part. Denote F_{19} the set of boundary faces of SDV[19], $F_{19} = \{ f_i \}, i = 1,\ 2,\ ...,\ 10$. That is

$$\begin{aligned} \text{SDV[19].closed\_face} &= \{ f_i \} \big| \ f_i \in F_{19} \wedge f_i \cap P_{\mathrm{B}} \neq \varnothing \\ &= \{ f_i \} \big| f_i \in F_{19},\ i = 1,2,...,5 \end{aligned} \tag{5.5}$$

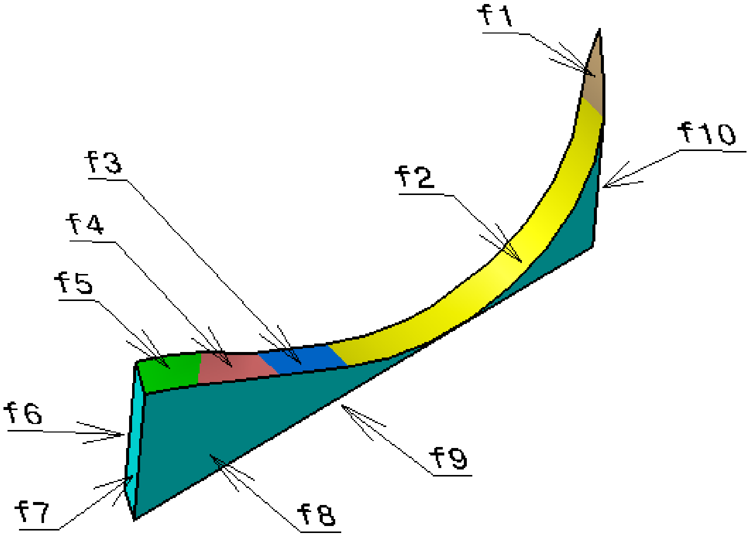


Figure 5-4 Boundary faces of SDV[19] (turned down for better view effect)

And apply equation (4.1) and (4.2) to SDV[19] we get

$$\begin{aligned} \text{SDV[19].f\_acc} = \text{SDV[19].open\_face} &= F_{19} -^{*} \text{SDV[19].closed\_face} \\ &= \{ f_i \} \big| f_i \in F_{19},\ i = 6,7,...,10 \end{aligned} \tag{5.6}$$

The set of open faces of SDV[21] and SDV[24] can be determined similarly.

5.2 Access point

As described in section 4.2.2, each access face has a corresponding access point. An access point can be the geometrical center of the face, the middle point of a non-part edge as well as a non-part vertex. It can also be any interior point of a non-part edge. The complete set of access points of a SDV is defined by equation (4.11).

As an example we do not determine the complete set of access point of the SDV. In the following examples we are going to find out only the geometrical centers of access faces.

First take SDV[4] for example. As the set of open faces of SDV[4] has been determined by equation (5.3), apply equation (4.6) to the set SDV[4].open_face, we get

$$\begin{aligned}\text{SDV[4].geoc\_openf} &= \mathit{get\_geocenter}(\text{SDV[4].open\_face}) \\ &= \{c_1, c_2, c_4, c_5, c_6\}\end{aligned} \tag{5.7}$$

where c_i is the geometrical center of $f_i \in \text{SDV[4].f\_acc}$. As we consider only the geometrical centers of the open faces in the example, according to (4.8) we have

$$\text{SDV[4].p\_acc} = \{c_1, c_2, c_4, c_5, c_6\} \tag{5.8}$$

The access points of SDV[4] are illustrated in Figure 5-5.

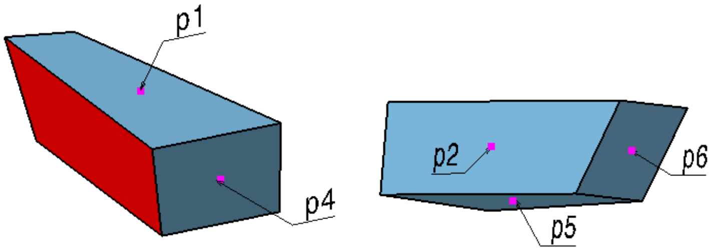


Figure 5-5 Access points of SDV[4]

Similarly we can determine the set of access points of SDV[19], SDV[21] and SDV[24] as illustrated in Figure 5-6.

5.3 Tool axis vectors

As described in section 4.5 an access vector, which represents the tool axis direction, can be associated with each access point. The access vector can be parallel to Z axis, perpendicular to the access face or the corresponding bottom face. If there is a folded face or shielded face, as in the case of SD[19], more steps have to be taken to determine the tool axis vector. In this section the approaches developed in section 4.5 will be applied to determine the set of tool vectors of each sample SDV.

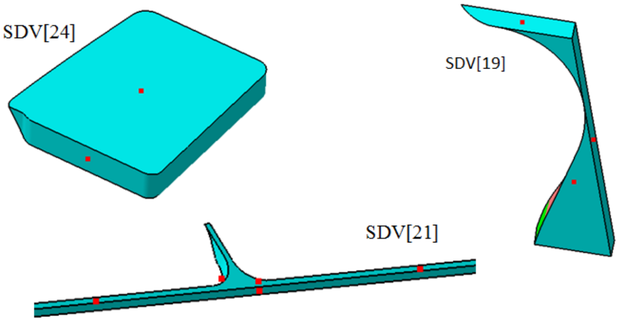


Figure 5-6 Access points of SDV[19], SDV[21] and SDV[24]

5.3.1 Tool axis vector of SDV[4]

Figure 5-7 shows the normal vectors of the open faces of SDV[4].

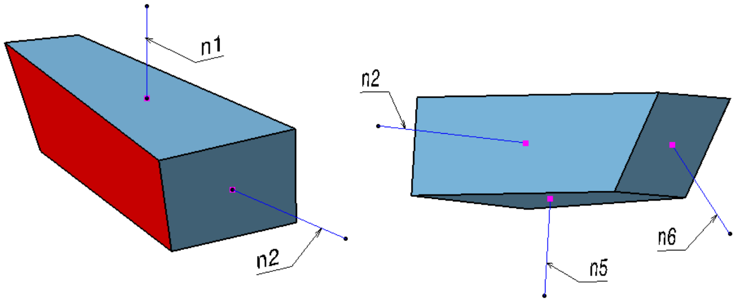


Figure 5-7 Normal vectors of the open faces of SDV[4]

Denote $\mathbf{V}_{n4} = \{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_4, \mathbf{n}_5, \mathbf{n}_6\}$ the set of normal vectors of the open faces of SDV[4], apply principles described in section 4.5.1 to $\mathbf{V}_{n4}$ and SDV[4].f_acc, we get the result that f_1 is the face, actually the only face of the set that satisfies equations (4.47) and (4.48), shown in Figure 5-7. That is

$$\mathbf{w}\mathbf{n}_1 > 0 \tag{5.9}$$

and

$$\mathrm{prj}f_1 \supseteq \bigcup \mathrm{prj}f_i \,|\, f_1,\ f_i \in \mathrm{SDV[4].f\_acc} \tag{5.10}$$

Denote $\mathbf{s}_4$ the set of tool axis vectors of SDV[4], we have

$$\mathbf{s}_4 = \{\mathbf{n}_1\} = \{\mathbf{w}\} \tag{5.11}$$

Which means to machine SDV[4] starting from face f_1 the tool can be initially oriented parallel to z axis.

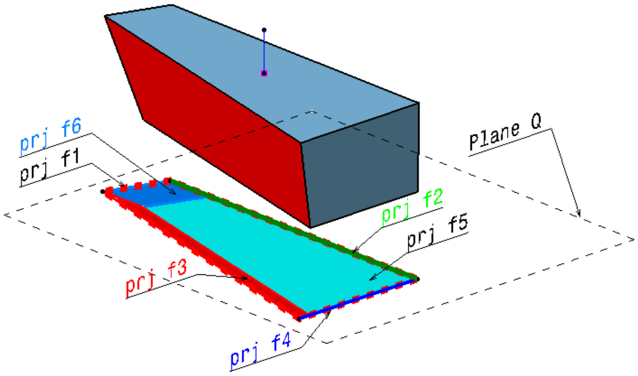


Figure 5-8 Projections of the boundary faces of SDV[4]

Similarly the tool axis vectors of SDV[21] and SDV[24] are determined parallel to z-axis as shown in Figure 5-9.

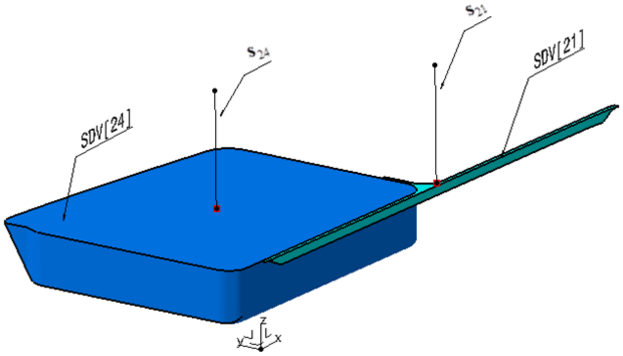


Figure 5-9 Tool axis vectors of SDV[21 and SDV[24]

5.3.2 Tool axis vectors of SDV[19]

By verifying it is found that SDV[19] does not belong any of the categories discussed in section 4.5.1 through 4.5.4. Then this SDV is checked by applying approaches introduced in section

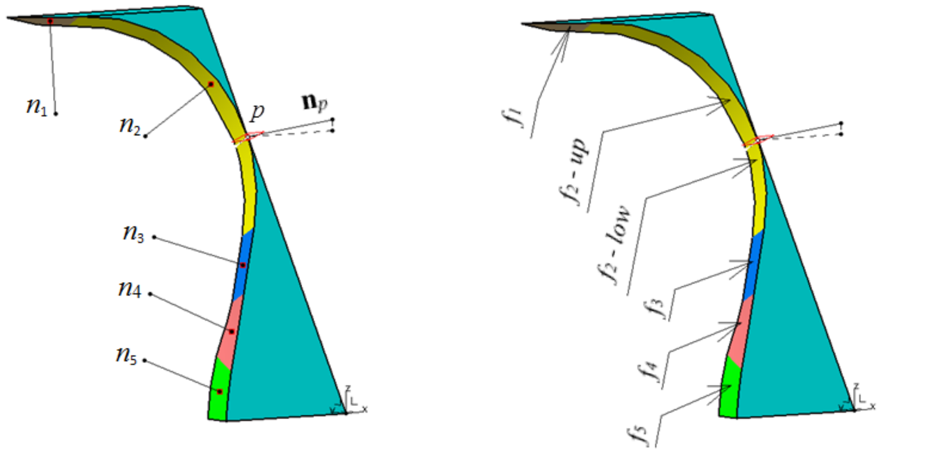
4.5.5 to detect the folded face in its boundary faces. First determine vectors $\{\mathbf{n}_i\}, i=1,2,...,5$ normal to the part faces $\{f_i\}, i=1,2,...,5$ of the SDV at their geometrical centers respectively. From Figure 5-10 (a) we know directly that the normal vectors of the part faces of SDV[19] , determined by (5.5), have respectively z-components as follows

$$n_{z1},\ n_{z2} < 0 \text{ and } n_{z3},\ n_{z4},\ n_{z5} > 0$$

Apply criteria (4.53) to the z-components listed above, we get

$$\prod_{i=1}^{5} neg(n_{zi}) = 0 \tag{5.12}$$

Thus faces $\{f_i\}, i=1,...,5$ of SDV[19] form a folded face chain.



(a) Normal vectors of part faces

(b) Upper and lower sections of folded face chain

Figure 5-10 Normal vectors and curve *C* of part faces of SDV[19]

By applying equation (4.54) to each face f_j of $\{f_i\}, i=1,...,5$, the point p and curve C are detected in face f_2. It can see clearly from Figure 5-10 (b) that face f_2 is a folded face as viewed in direction $-\mathbf{w}$. Curve *C* separates the folded face chain into upper and lower sections as illustrated in Figure 5-10 (b).

According to the approaches given in section 4.5.6, the point in each face where the normal has maximum z-component has to be determined. This is done by applying equation (4.56) to $\{f_i\}, i=1,...,5$ of SDV[19], thus obtain M_i $(i=1,2,...,5)$. Figure 5-11 shows the points in the part faces where normals vectors denoted $N1, N2, ..., N5$ corresponding to M_i $(i = 1, 2, ..., 5)$ respectively are detected. As the result we have

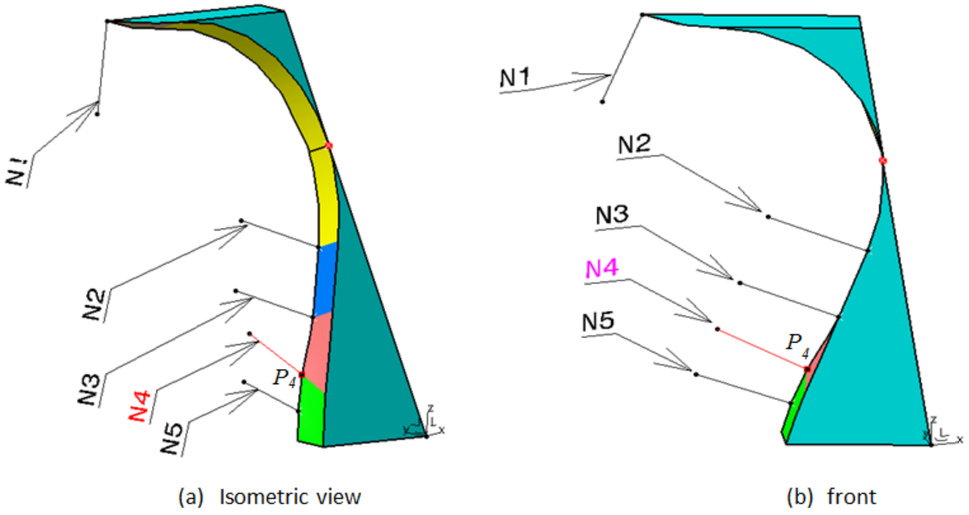
$$N4_z = M_4 = \max\left(\{M_i\}\right),\ i = 1, 2, ..., 5$$

Corresponding to M_4 face f_4 and point p_4 are detected. As we know in advanced B-rep each face is associated with its surface which is represented by an implicit function. The function for face f_4 is denoted F_4. Substitute F_j and q in equation (4.58) by surface F_4 and point p_4 respectively we get the normal vector $\mathbf{n}_{p_4}$ as following

$$\mathbf{n}_{p_4} = \left.\frac{\nabla F_4}{\|\nabla F_4\|}\right|_{p_4} \tag{5.13}$$

Determine planes P and Q in accordance to p_4 and $\mathbf{n}_{p_4}$, determine the intersection R of the two planes as discussed in section 4.5.6, the results are illustrated in Figure 5-12(a). As having been mentioned at the beginning of section 4.5, a tool axis vector can be determined by

In all directions accessible to all part faces of the SDV, has the minimum deflection angle from z-axis. For better view effects in a tool is positioned at point p_4 with radial compensation as shown in Figure 5-12(b). In practice a tool must be positioned at the access point while starting machine a SDV.



(a) Isometric view

(b) front

Figure 5-11 Normal vectors of part faces of SDV[19]

5.4 Length of cutting edge

To verify the approaches discussed in section 4.9.1 we are going to determine the length of cutting edges corresponding to SDV[19] and SDV[21] respectively.

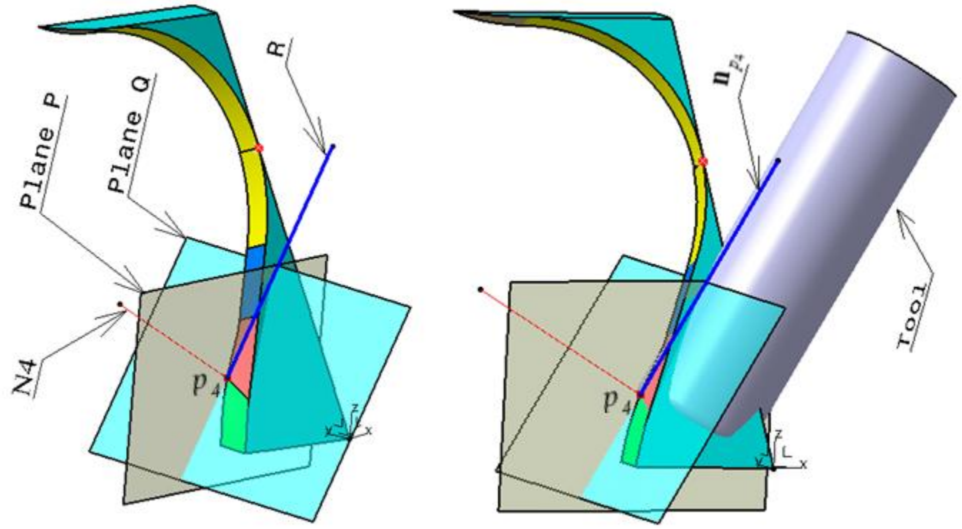
First, the point *p* in the boundary of SDV[19] which is farthest from the access point as measured in corresponding tool axis vector **s** has to be detected. This purpose can be achieved by

$$p = extrem(\text{SDV[19].f}, -\mathbf{s}) \tag{5.14}$$

Second, create ray L passing *p* and parallel to **s**.

$$\mathrm{L} = ray(p, \mathbf{s}) \tag{5.15}$$

Where $ray(p,\mathbf{s})$ is the function create a ray passing *p* and parallel to **s** as shown in Fig 5.13.



(a) Plane *P*, *Q* and intersection *R*

(b) Vector $\mathbf{n}_{p_4}$ and tool position

Figure 5-12 Tool axis vector of SDV[19]

Third, create plane *Q* passing *p* and has its normal determined by equation (4.59).

Fourth, determine the intersection of L and the boundary faces of the final stock

$$q = \mathrm{L} \cap \text{SF.f} \tag{5.16}$$

As we know that L is in plane *Q*, thus equation (5.16) is a special case of the situations determined by equations (4.81) and (4.82).

Finally the distance between *p* and *q* is the minimum length of the tool used to machine SDV[19].

In Fig 5.13 the tool axis vector **s** is shown at the access point which is the middle point of an edge bounding the top face of the SDV.

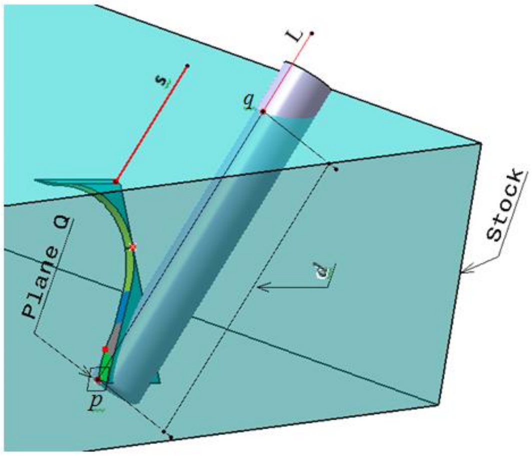


Figure 5-13 Minimum length of tool for machining SDV[19]

In Figure 5-14 the center portion of SDV[21] is shown in multi-view. From views (b), (c) and (d) we know that the top and bottom faces are parallel planar faces. According to discussion made in section 4.9.1, point *p* is the projection of p_{acc} onto plane *Q* which is in this case overlapped with the bottom face.

Similar to what we did to SDV[19] determine point *q* the intersection of ray L and the boundary of the stock denoted by SF.f, see Figure 5-14. Then the distance *d* between *p* and *q* is the minimum length of the tool used to machine SDV[21], as shown in Figure 5-15.

The minimum tool length for SDV[4] and SDV[24] can be determined similarly.

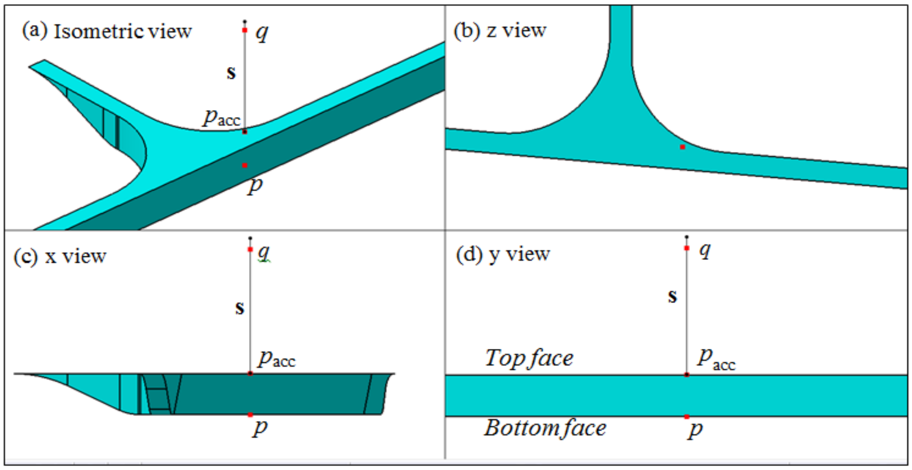


Figure 5-14 Multi view of SDV[21]

5.5 Diameter of tool

As discussed in section 4.9.2 under the circumstance that the criteria (4.91) is satisfied, then the dimension of diameter of a tool is limited by the part faces between the access face and corresponding bottom face.

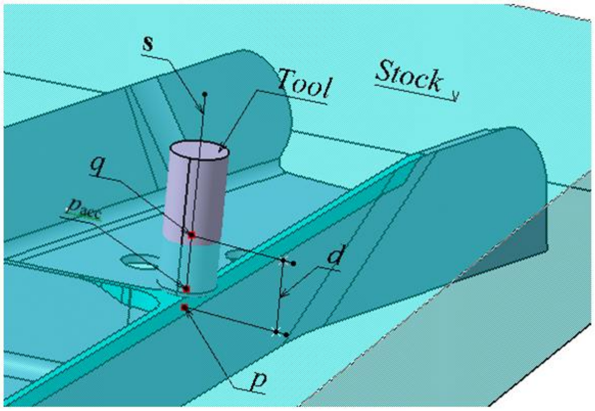


Figure 5-15 Minimum tool length for SDV[21]

After verifying SDV[4], SDV[19] and SDV[21] it is found that no faces satisfy the criteria (4.91). Thus for these SDVs there is no limit on the diameter of tool.

In the boundary of SDV[24] between the top face and bottom face there are two part faces f_j and f_k satisfy criteria (4.91) that is

$$\mathbf{n}_{Qj}\mathbf{n}_{Qk} < 0 \tag{5.17}$$

as illustrated in Figure 5-16.

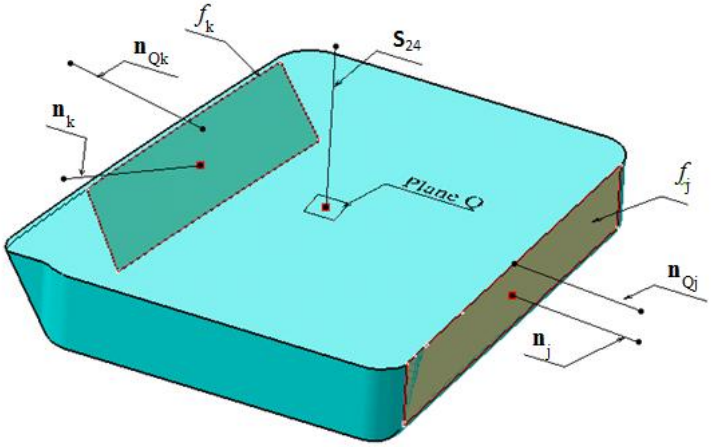


Figure 5-16 Tool diameter for SDV[24]

For better view effects in Figure 5-17 projections of faces f_j and f_k onto plane *Q*, denoted f_{Qj} and f_{Qk} respectively, are illustrated. The distances between these two faces and the line L are the same as the distances d_j and d_k which is the distances between point p_{acc} and the projections respectively.

According to equation (4.85) as d_k is the smallest distance between L and part faces of SDV[24], the tool diameter *D* is limited by

$$\frac{D}{2} \le d_k \tag{5.18}$$

as illustrated in Figure 5-17.

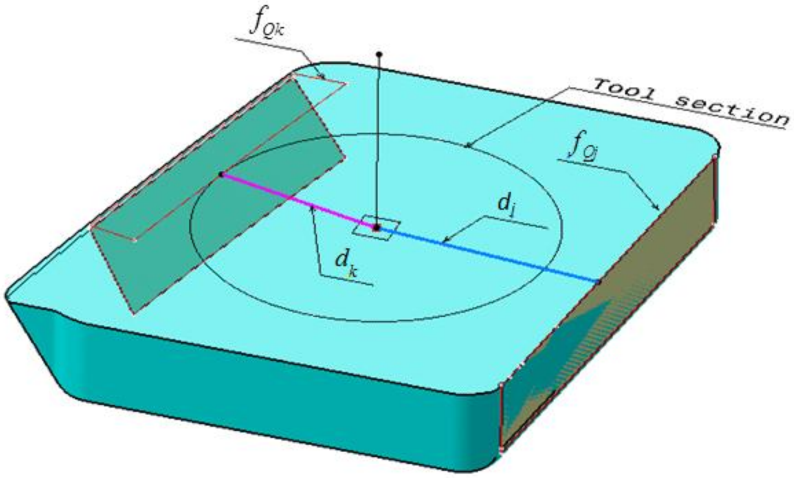


Figure 5-17 Tool diameter for SDV[24]

5.6 Type of tool tip and operation assigning

As discussed in section 4.9.3 the type of tool tip is determined by both the direction and curvature of the part face to be machined. Sub delta volume SDV[24] is taken as an example to verify the proposed principles.

Figure 5-18 (a) shows SDV[24] and the offsets of some part faces. Sample faces f_1, f_2, f_3, f_4 and f_5, shown in Figure 5-18 (b) will be verified. In Figure 5-18 (c), the normal vectors of these sample faces are shown as $\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3, \mathbf{n}_4$ and $\mathbf{n}_5$, for better view effects, the access face is illustrated by its offset. Vector $\mathbf{s}_{24}$ is the tool axis vector corresponding to the access face.

For the bottom face f_1, as $\mathbf{n}_1\mathbf{s}_{24} = -1$, thus a flat end mill is preferred tool to machine this face from $\mathbf{s}_{24}$.

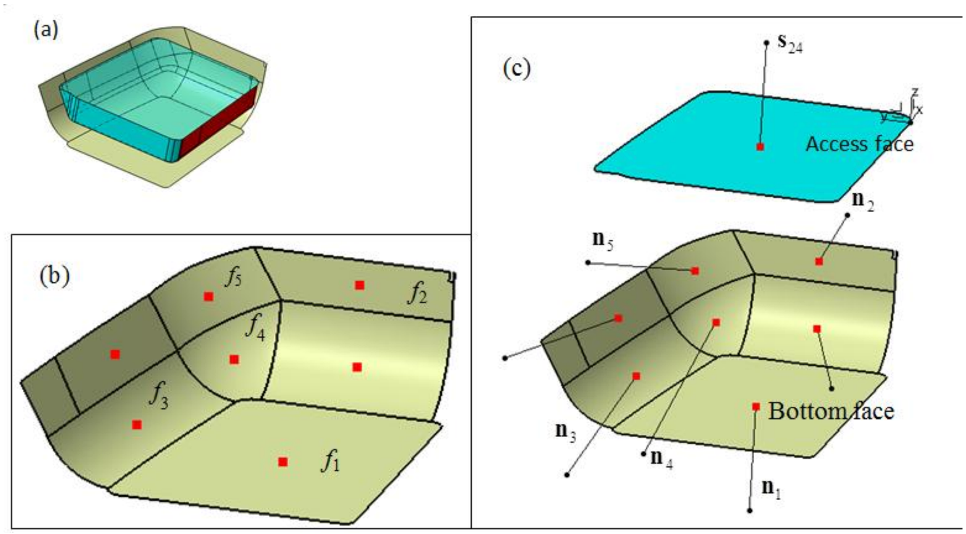


Figure 5-18 Faces and normal vectors of SDV[24]

As $\mathbf{n}_2\mathbf{s}_{24} = 0$, face f_2 does not limit the tool tip. Both faces f_3 and f_5 have one dimension curvature and $-1 < \mathbf{n}_3\mathbf{s}_{24} < 0$, $-1 < \mathbf{n}_5\mathbf{s}_{24} = 0$, thus a ball end mill is required to machine these two faces from direction $\mathbf{s}_{24}$.

Face f_4 has two dimension curvatures, so a ball end mill must be used to machine it.

As discussed in section 4.9.3, faces sharing edges with a face requiring a ball end mill, these adjacent faces have to be machined with this ball end mill in areas neighbouring to the common edges. In this example, as faces f_1, f_2, f_3, and f_5 share common edges with face f_4, in the neighbour areas of the common edges, these faces have to be machined with the ball end mill determined by face f_4.

To machine SDV[24] in one operation, a ball end mill must be selected. As have been mentioned in section 4.9.3, using tools of different tips may increase machining efficiency, and this issue is left for future researches.

Apply the approaches developed in section 4.8 to SDV[24], as $J_{op} = 2$ and $J_p > 2$ in this case, and there are no open faces in the boundary of SDV[24] have opposite direction, thus a pocket operation is assigned to the SDV.

It is obviously noticed that SDV[4], SDV[19] and SDV[21] satisfy the criteria of contouring operation. For example, to SDV[4], $N = 1$, it can be machined by contouring operation. To SDV[19] and SDV[21], $\prod \mathbf{n}_i > 0$ is satisfied, and every has two non-adjacent edges shared by two non-adjacent open faces respectively, it has two open faces adjacent with the same , thus they are assigned contouring operations respectively.

5.7 Operation sequencing

As discussed in section 4.10, this research proposed approaches to generating operation sequence is realized by ordering all elements of the set SDVs, because every SDV determines the space area that the effective part of a tool selected correspondingly to the SDV can sweep all over (with compensation to the s), and the machining attributes associated with the SDV carry manufacturing information about with what and how will the SDV be removed. In this section some SDVs of part II will be taken as examples to illustrate the application of the proposed approaches.

Facing has the first priority as proposed in section 4.10, thus SDV[1] the facing SDV gets the first in the ordered list.

Interior SDVs accessible from z-axis has the second priority. In our example, $\text{SDV}[i], i \geq 19$ are interior SDVs. Verify the part faces of these SDVs with criteria (4.80) , we get elements of G$_{\text{pos}}$. Some of the elements are shown in Figure 5-19 with their geometric centers highlighted by small red squares. Figure 5-19 (a) shows the original position of the SDVs. In Figure 5-19 (b) SDV[1] the facing SDV is moved to a higher position, the rests remain their original positions.

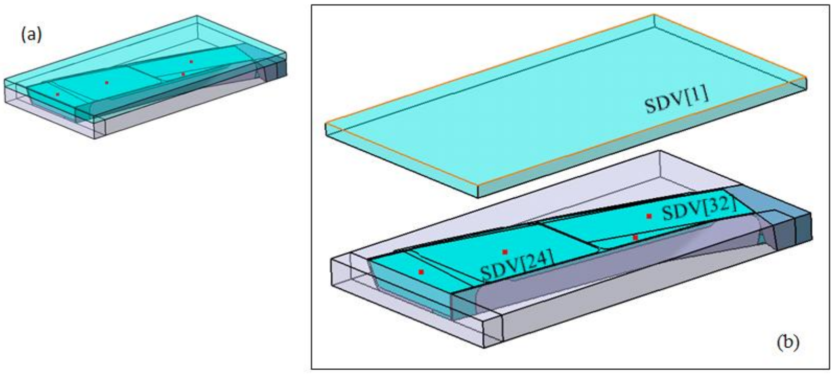


Figure 5-19 Facing SDV and interior SDVs of Part II

To sequence these highlighted interior SDVs, as they are the same height, the fifth criteria (area of top face priority) is considered. As SDV[24] is larger than SDV[32], it is listed before. Denote $\{sq_i\}$ the list of sequence, till now we have $sq_1 = \text{SDV[1]}$, $sq_2 = \text{SDV[24]}$ and $sq_3 = \text{SDV[32]}$. Continue put the rest highlighted SDVs in the list according to the areas of their top faces. In Figure 5-20 the procedure of putting a SDV in the list were illustrated by hiding the SDV each time when it is put in the operation list, and the result was shown in the next window in the figure.

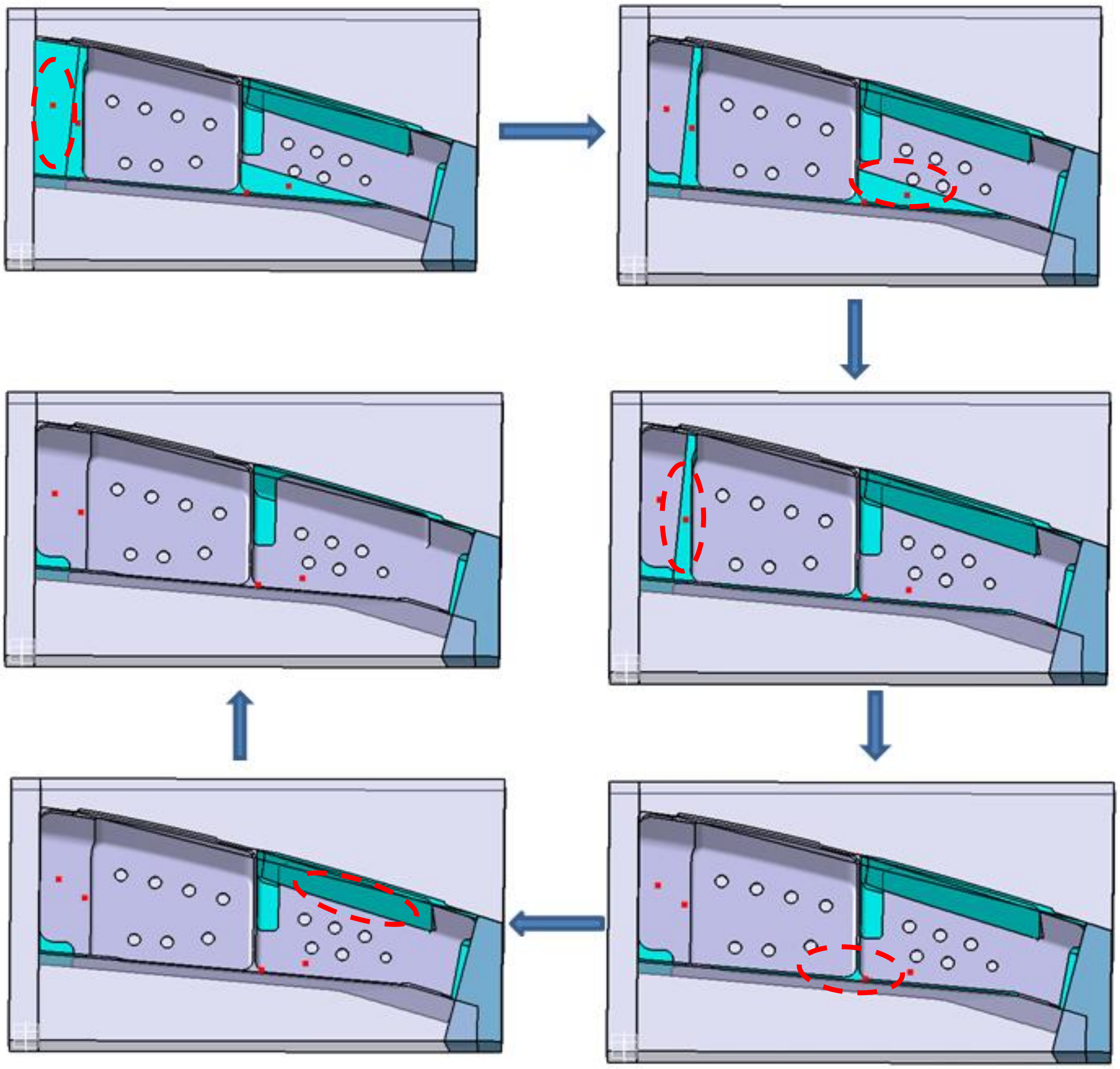


Figure 5-20 Procedures of putting the SDVs in the operation list

After putting all SDVs of the same height as SDV[24] and being accessible from z-axis into the operation list, we consider the SDVs in the set G_{piv} (SDVs accessible from a direction pivoting from z). Figure 5-21 illustrates the processes of putting these SDVs in the operation list. SDV[19] is the first of the SDVs in G_{piv} being put in the list according to the eighth priority proposed in section 4.10. For better view effects, the exterior SDVs were hidden.

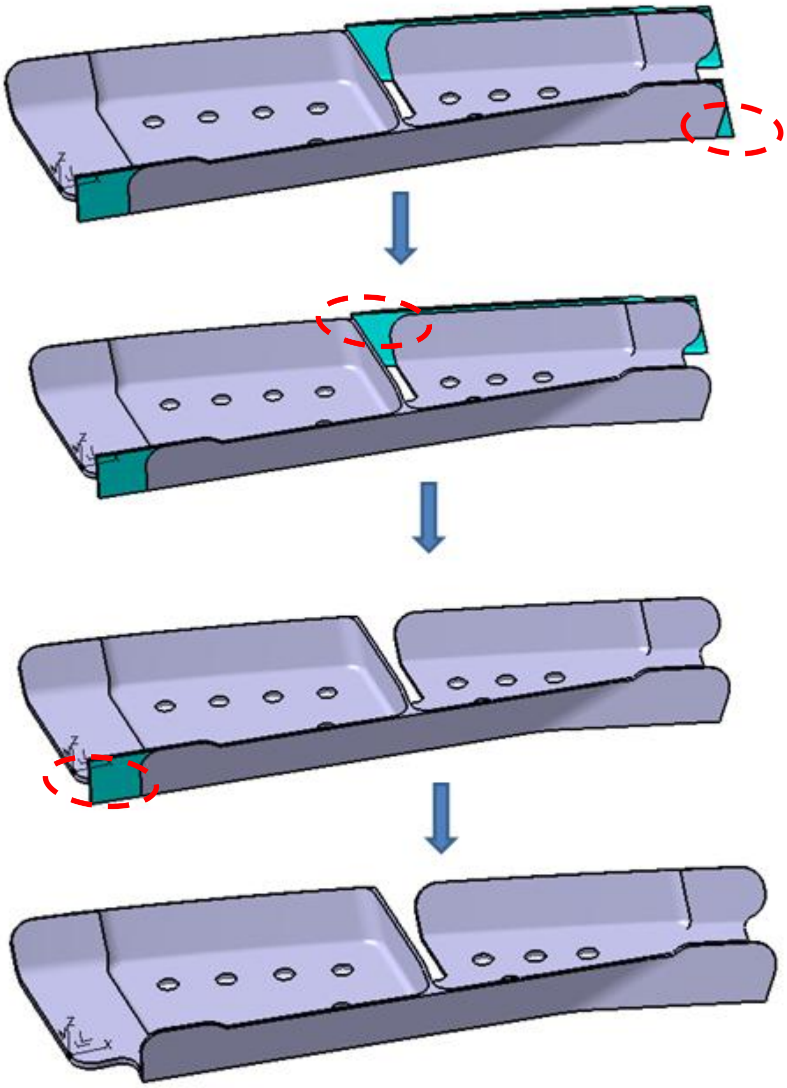


Figure 5-21 Processes of putting SDVs of set G_{piv} into the operation list

Till now all interior SDVs have been putting in the operation list. The rest are all exterior SDVs. To save space the processes of putting the exterior SDVs in the operation list are omitted.

CONCLUSION AND RECOMMENDATIONS

The main objective of the present research project was to develop a robust systematic approaches to transferring the CAD model of an aeronautical structural part into a CAM model, which includes the following information: 1) the geometric definition of removable volumes of material being able to be machined in one machining operation; 2) machining attributes associated with the geometry representations of individual volumes correspondingly; 3) machining operations assigned to individual volumes respectively; and 4) list of ordered operations. Generating separating faces was the key step to conduct the delta volume decomposition by Boolean operation. Approaches to creating different categories of separating faces were developed according to a complex aeronautical structural part. Distinguishing open and s of individual volumes, comprehensive analysis on the geometries and related special positions of the volumes make the way to the purposes of associating machining attributes with the removable volume of material, assigning machining operations to the volumes. Establishing priorities for ordering the volumes in a list so that generate machining operations sequence was accomplished.

With the proposed approaches for delta volume decomposition, the representation of a complex aeronautic structural component can be converted from its CAD model to a corresponding CAM model using a set of sub delta volumes. Each element of the set represents a volume of material that is removable in one milling/drilling operation. Detection of splitting location, identification of the front/back concave vertex, distinguishing exterior/interior SDVs, and application of face union in decomposition are some characteristics of the proposed decomposition methods. The proposed approaches also provide functions to expand a stock to reasonable dimensions as required by an inclined hole, to renew the models of related SDVs.

The proposed approaches analyze machining attributes of an SDV corresponding to each of its open faces as well as s. This allows the user to find all possible access faces and corresponding tool parameters, which is useful information for optimization analysis. Distinguishing between angular and linear accessibilities facilitates the analysis of machinability. By taking the existence of other SDVs into account, the proposed approaches are applicable when alternative operation sequences should be generated.

As a result of applying the proposed approaches, we can know whether a part is machinable or not, the required number of axis of a machine, how many different tools are needed to produce the part, how many machining operations are suggested, etc.

The present thesis is the first that propose the integration of approaches to converting the CAD model to CAM model on the base of sub delta volume.

The main contributions of this research can be summarized as following:

1) Approaches to integrating delta volume decomposition, machining attributes determination, machining operation assignment and operation sequencing;
2) Approaches to creating faces for separating sub delta volumes, creating face union and applying it to conduct separation;
3) Concepts and approaches to identifying closed and open faces;
4) Identifying and associating all possible access positions to a SDV;
5) Approaches to analyzing inclined hole and updating the rough stock and adjacent SDVs;
6) Mathematical definition and approaches to identify concave edges;
7) Concepts of required tool posture flat and space, and approaches to verifying the machinability;
8) Approached to determining tool parameters;
9) Proposed priorities of sub delta volume sequencing.

The example showing the complete procedures of the model conversion confirms the applicability of the proposed approaches.

Considering the limitation of the research, some recommendations and avenues for future work could be investigated to provide some answers to the points raised in this research:

1. The impact of a tool on other SDVs while machining a destination SDV;
2. Approaches to avoiding a SDV have an extruded volume on the side face(s);
3. Approaches to selecting the access direction from which a SDV can be machined most efficiently;
4. Approaches to decomposing an existing SDV to gain machining efficiency;

5. Surveying the necessity of assigning multiple tools to one SDV for roughing;
6. Associating important factors, such as material, machining parameters, tooling fixtures and tolerances with the SDVs;
7. The proposed approaches do not have the capability to merge two or more adjacent SDVs into one so that they can be machined in one operation. There is the possibility that combining some adjacent SDVs can improve machining efficiency, decrease the required number of tools and reduce manufacturing cost.
8. This research proposes some considerations for operation sequencing, but in fact other better proposals for sequencing may exist. The development of methods for sequencing operations to achieve objectives such as shortest machining time and lowest manufacturing cost is an interesting subject for further study.
9. Approaches to changing tool parameters according to the change of the operation sequence, which can be made either by a process programmer or an application for operation optimizations.
10. There is still much work to do to implement the proposed approaches in a real CAD/CAM system.

From a manufacturing point of view, it is of great interest to determine whether a machining operation sequence is an optimization one. The results of this study provide the conveniences to optimize the operation sequence. It is recommended to integrate the proposed approaches with optimization applications.

REFERENCES

[1] C. M. d. Montréal, "Aerospace Cluster," p. 7, 2004.

[2] G. o. Canada, "Canada as an investment destination for aerospace," ed.

[3] P. J. Clarkson, C. Simons, and C. Eckert, "Predicting change propagation in complex design," *Journal of Mechanical Design, Transactions of the ASME*, vol. 126, pp. 788-797, 2004.

[4] J. Corney, C. Hayes, V. Sundararajan, and P. Wright, "The CAD/CAM interface: A 25-year retrospective," *Journal of Computing and Information Science in Engineering*, vol. 5, pp. 188-197, Sep 2005.

[5] A. Oral and M. C. Cakir, "Automated cutting tool selection and cutting tool sequence optimisation for rotational parts," *Robotics and Computer-Integrated Manufacturing*, vol. 20, pp. 127-141, Apr 2004.

[6] H. F. Wang and Y. L. Zhang, "CAD/CAM integrated system in collaborative development environment," *Robotics and Computer-Integrated Manufacturing*, vol. 18, pp. 135-145, Apr 2002.

[7] K. J. Fernandes and V. H. Raja, "Incorporated tool selection system using object technology," *International Journal of Machine Tools & Manufacture*, vol. 40, pp. 1547-1555, Sep 2000.

[8] E. Dimas and D. Briassoulis, "3D geometric modelling based on NURBS: a review," *Advances in Engineering Software*, vol. 30, pp. 741-751, 1999.

[9] M. Mantyla, D. Nau, and J. Shah, "Challenges in feature-based manufacturing research," *Communications of the ACM*, vol. 39, pp. 77-85, 1996.

[10] N. Sridharan and J. J. Shah, "Recognition of multi axis milling features: Part I - Topological and geometric characteristics," *Journal of Computing and Information Science in Engineering*, vol. 4, pp. 242-250, Sep 2004.

[11] J. H. Han, M. Pratt, and W. C. Regli, "Manufacturing feature recognition from solid models: A status report," *Ieee Transactions on Robotics and Automation*, vol. 16, pp. 782-796, Dec 2000.

[12] N. Sridharan and J. J. Shah, "Recognition of multi-axis milling features: Part II - Algorithms & implementation," *Journal of Computing and Information Science in Engineering*, vol. 5, pp. 25-34, Mar 2005.

[13] C. M. Hoffmann and R. Joan-Arinyo, "On user-defined features," *Computer-Aided Design*, vol. 30, pp. 321-332, 1998.

[14] J. H. Vandenbrande and A. A. G. Requicha, "Spatial reasoning for the automatic recognition of machinable features in solid models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1269-1285, 1993.

[15] I. C. Braid and C. A. Lang, "Computer-aided design of mechanical components with volume building bricks," *Automatica*, vol. 10, pp. 635-42, 1974 1974.

[16] I. C. Braid and C. A. Lang, "COMPUTER-AIDED DESIGN OF MECHANICAL COMPONENTS WITH VOLUME BUILDING BRICKS," *Automatica,* vol. 10, pp. 635-642, 1974.

[17] L. K. Kyprianou, "Shape classification in computer-aided design," CA92850, University of Cambridge (United Kingdom), England, 1980.

[18] W. C. Regli and D. S. Nau, "Building a general approach to feature recognition of material removal shape element volumes (MRSEVs)," in *Proceedings of the 2nd Symposium on Solid Modeling and Applications,* pp. 293-302, 1993.

[19] S. Finger and S. A. Safier, "Representing and recognizing features in mechanical designs," in *1990 ASME Design Technical Conferences - Presented at the 2nd International Conference on Design Theory and Methodology*, pp. 19-25, 1990.

[20] A. A. G. Requicha and J. H. Vandenbrande, "Form features for mechanical design and manufacturing," in *Proceedings of the 1989 ASME International Computers in Engineering Conference and Exposition*, pp. 47-52, 1989.

[21] J. H. Vandenbrande, "Automatic recognition of machinable features in solid models," 9111513, The University of Rochester, United States -- New York, 1990.

[22] S. Venkatadriagaram, "Feature recognition and high-level process planning for rapid prototyping using 3-axis milling," 3002302, University of California, Berkeley, United States -- California, 2000.

[23] Y. Woo and H. Sakurai, "Recognition of maximal features by volume decomposition," *Computer-Aided Design,* vol. 34, pp. 195-207, Mar 2002.

[24] W. C. Regli, S. K. Gupta, and D. S. Nau, "Towards multiprocessor feature recognition," *Computer-Aided Design,* vol. 29, pp. 37-51, Jan 1997.

[25] V. Sundararajan and P. K. Wright, "Volumetric feature recognition for machining components with freeform surfaces," *CAD Computer Aided Design,* vol. 36, pp. 11-25, 2004.

[26] T. Lim, H. Medellin, J. R. Corney, J. M. Ritchie, and J. B. C. Davies, "Decomposition of complex models for manufacturing," in *Proceedings - Shape Modeling International SMI*, 2004, pp. 337-341.

[27] E. Van den Berg, W. F. Bronsvoort, and J. S. M. Vergeest, "Freeform feature modelling: concepts and prospects," *Computers in Industry,* vol. 49, pp. 217-233, 2002.

[28] M. G. Marchetta and R. Q. Forradellas, "An artificial intelligence planning approach to manufacturing feature recognition," *Computer-Aided Design,* vol. 42, pp. 248-256, 2010.

[29] P. Di Stefano, F. Bianconi, and L. Di Angelo, "An approach for feature semantics recognition in geometric models," *CAD Computer Aided Design,* vol. 36, pp. 993-1009, 2004.

[30] S. H. Chuang and M. R. Henderson, "Three-dimensional shape pattern recognition using vertex classification and vertex-edge graphs," *Computer-Aided Design,* vol. 22, pp. 377-387, 1990.

[31] S.-H. Chuang and M. R. Henderson, "Compound feature recognition by web grammar parsing," *Research in Engineering Design,* vol. 2, pp. 147-58, 199.

[32] J. K. Coles, R. H. Crawford, and K. L. Wood, "Form feature recognition using base volume decomposition," in *Proceedings of the 1994 ASME Design Technical Conferences. Part 1 (of 2),* pp. 281-297, 1994.

[33] M. Flasinski, "Use of graph grammars for the description of mechanical parts," *Computer-Aided Design,* vol. 27, pp. 403-433, 1995.

[34] M. Flasinski, "Characteristics of edNLC-graph grammar for syntactic pattern recognition," *Computer Vision, Graphics, and Image Processing,* vol. 47, pp. 1-21, 1989.

[35] D. Ben-Arieh, "Geometrical reasoning based on attributed graph grammar for prismatic parts," *IIE Transactions (Institute of Industrial Engineers),* vol. 31, pp. 61-74, 1999 1999.

[36] J. Han, W. C. Regli, and S. Brooks, "Hint-based reasoning for feature recognition: status report," *Computer-Aided Design,* vol. 30, pp. 1003-1007, 1998.

[37] W. C. Regli, S. K. Gupta, and D. S. Nau, "Extracting alternative machining features: an algorithmic approach," *Research in Engineering Design - Theory, Applications, and Concurrent Engineering,* vol. 7, pp. 173-192, 1995.

[38] J. Han and A. A. G. Requicha, "Integration of feature based design and feature recognition," *Computer-Aided Design,* vol. 29, pp. 393-403, 1997.

[39] J. H. Han, W. C. Regli, and S. Brooks, "Hint-based reasoning for feature recognition: status report," *Computer-Aided Design,* vol. 30, pp. 1003-1007, Nov 1998.

[40] K. Rahmani and B. Arezoo, "A hybrid hint-based and graph-based framework for recognition of interacting milling features," *Computers in Industry,* vol. In Press, Corrected Proof, 2006.

[41] T. C. Woo, "Feature Extraction by Volume Decomposition," *Proc. Conf. CAD/CAM Technology in Mechanical Engineering,* pp. 76-94, 1982.

[42] D. L. Waco and Y. S. Kim, "Geometric reasoning for machining features using convex decomposition," in *Proceedings of the 2nd Symposium on Solid Modeling and Applications,* pp. 323-332, 1993.

[43] A. G. a. R. J. G. H. B. Marri, "Computer-aided process planning: A state of art," *The International Journal of Advanced Manufacturing Technology* vol. 4, pp. 261-268, 1998.

[44] Y. S. Kim, Y. Kim, F. Pariente, and E. Wang, "Geometric reasoning for mill-turn machining process planning," *Computers & Industrial Engineering,* vol. 33, pp. 501-504, 1997.

[45] H. Sakurai and P. Dave, "Volume decomposition and feature recognition, part II: Curved objects," *CAD Computer Aided Design,* vol. 28, pp. 519-537, 1996.

[46] H. Sakurai, "Volume decomposition and feature recognition. I: Polyhedral objects," *Computer Aided Design,* vol. 27, pp. 833-843, 1995.

[47] H. Sakurai and C. W. Chin, "Form feature recognition by spatial decomposition and composition," in *Geometric Modeling for Product Realization. IFIP TC5/WG5.2 Working*

Conference on Geometric Modeling, 27 Sept.-1 Oct. 1993: IFIP Transactions B (Applications in Technology), 1992, pp. 189-203.

[48] I. E. Sutherland, "Sketchpad man-machine graphical communication system," in *American Federation of Information Processing Societies -- Joint Computer Conference*, 1963, pp. 329-346.

[49] M. T. Wang and T. C. Chang, "Feature recognition for automated process planning," in *Proceedings of Manufacturing International '90. Part 2: Advances in Manufacturing Systems*, 1990, pp. 49-54.

[50] Z. Gu, Y. F. Zhang, and A. Y. C. Nee, "Identification of important features for machining operations sequence generation," *International Journal of Production Research*, vol. 35, pp. 2285-2308, 1997.

[51] J. Ciurana, M. L. Garcia-Romeu, I. Ferrer, and M. Casadesús, "A model for integrating process planning and production planning and control in machining processes," *Robotics and Computer-Integrated Manufacturing*, vol. 24, pp. 532-544, 2008.

[52] A. V. S. R. K. Prasad, P. N. Rao, and U. R. K. Rao, "Optimal selection of process parameters for turning operations in a CAPP system," *International Journal of Production Research*, vol. 35, pp. 1495-1522, 1997.

[53] M. C. Kayacan, I. H. Filiz, and e. al, "OPPS-ROT: an optimised process planning system for rotational parts," *Computers in Industry*, vol. 32, pp. 181-195, 1996.

[54] T. Dereli, I. H. Filiz, and e. al, "Optimizing cutting parameters in process planning of prismatic parts by using genetic algorithms," *International Journal of Production Research*, vol. 39, pp. 3303-3328, 2001.

[55] R. Siva Sankar, P. Asokan, G. Prabhaharan, and A. V. Phani, "A CAPP framework with optimized process parameters for rotational components," *International Journal of Production Research*, vol. 46, pp. 5561-5587, 2008.

[56] J. P. Kruth, G. Van Zeir, and J. Detand, "Extracting process planning information from various wire frame and feature based CAD systems," *Computers in Industry*, vol. 30, pp. 145-162, 1996.

[57] P. Gu and Y. Zhang, "Operation sequencing in an automated process planning system," *Journal of Intelligent Manufacturing*, vol. 4, pp. 219-32, 1993.

[58] M. Deja and M. Siemiatkowski, "Feature-based generation of machining process plans for optimised parts manufacture," *Journal of Intelligent Manufacturing*, pp. 1-16, 2012.

[59] G. Nallakumarasamy, P. S. S. Srinivasan, K. Venkatesh Raja, and R. Malayalamurthi, "Optimization of operation sequencing in CAPP using simulated annealing technique (SAT)," *The International Journal of Advanced Manufacturing Technology* vol. 54, pp. 751-728, 2011.

[60] J. D. Wang, B. Q.; Ding, H., "A Modified Genetic Algorithm(GA) for Optimization of Process Planning," *Journal of Computers*, vol. 6, pp. 1430-1437, 2011.

[61] C. Dartigues, P. Ghodous, M. Gruninger, D. Pallez, and R. Sriram, "CAD/CAPP integration using feature ontology," *Concurrent Engineering: Research and Applications*, vol. 15, pp. 237-49, 2007.

[62] K. N. Brown and J. Cagan, "Optimized process planning by generative simulated annealing," *(AI EDAM) Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 11, pp. 219-35, 1997.

[63] L.-H. Qiao, Z.-B. Yang, and H.-P. Ben Wang, "A computer-aided process planning methodology," *Computers in Industry,* vol. 25, pp. 83-94, 1994.

[64] F. L. Chen, D. Joo, and J. T. Black, "Machine vision in the automated detection and diagnosis of dimensional errors in end machining," *Computers & Industrial Engineering,* vol. 26, pp. 223-235, 1994.

[65] J. X. Gao and X. X. Huang, "Product and manufacturing capability modelling in an integrated CAD/process planning environment," *International Journal of Advanced Manufacturing Technology,* vol. 11, pp. 43-51, 1996.

[66] P. G. Maropoulos and R. P. Baker, "Integration of tool selection with design: Part 1. Feature creation and selection of operations and tools," *Journal of Materials Processing Technology,* vol. 107, pp. 127-134, 2000/.

[67] D. N. Sormaz and B. Khoshnevis, "Process planning knowledge representation using an object-oriented data model," *International Journal of Computer Integrated Manufacturing,* vol. 10, pp. 92-104, 1997.

[68] J. M. Usher and K. J. Fernandes, "Object-oriented application of tool selection in dynamic process planning," *International Journal of Production Research,* vol. 37, pp. 2879-2894, 1999.

[69] J. M. Usher and K. J. Fernandes, "A two-phased approach to dynamic process planning: Proceedings of the 19th International Conference on Computers and Industrial Engineering," *Computers & Industrial Engineering,* vol. 31, pp. 173-176, 1996.

[70] A. I. Shabaka and H. A. Elmaraghy, "Generation of machine configurations based on product features," *International Journal of Computer Integrated Manufacturing,* vol. 20, pp. 355-369, 2007.

[71] D. M. Gaines, F. Castano, and C. C. Hayes, "MEDIATOR: A resource adaptive feature recognizer that intertwines feature extraction and manufacturing analysis," *Journal of Mechanical Design, Transactions of the ASME,* vol. 121, pp. 145-158, 1999.

[72] F. González and P. Rosado, "General and flexible methodology and architecture for CAPP: GF-CAPP system," *International Journal of Production Research,* vol. 41, pp. 2643-2662, 2003.

[73] J. Balakrishnan and C. H. Cheng, "Multi-period planning and uncertainty issues in cellular manufacturing: A review and future directions," *European Journal of Operational Research,* vol. In Press, Corrected Proof, 2006.

[74] S. K. Gupta and D. S. Nau, "Systematic approach to analysing the manufacturability of machined parts," *Computer-Aided Design,* vol. 27, pp. 323-342, 1995.

[75] T. C. Chang, *Expert Process Planning for Manufacturing*. New York: Addison-Wesley Publishing Company, Inc., 1990.

[76] S. K. Gupta, T. R. Kramer, D. S. Nau, W. C. Regli, and G. Zhang, "Building MRSEV models for CAM applications," *Advances in Engineering Software,* vol. 20, pp. 121-139, 1994.

[77] W. Kramer, "Automation of materials handling," *British Plastics and Rubber,* pp. 12-13, 1994.

[78] E. S. Abouel Nasr and A. K. Kamrani, "A new methodology for extracting manufacturing features from CAD system," *Computers &amp; Industrial Engineering,* vol. 51, pp. 389-415, 2006.

[79] I. M. Deiab and S. C. Veldhuis, "Off line simulation system of machining processes," *Journal of the Franklin Institute,* vol. 344, pp. 565-576, 2007.

[80] H. C. Lee, W. C. Jhee, and H.-S. Park, "Generative CAPP through projective feature recognition: Selected Papers from The 27th. International Conference on Computers & Industrial Engineering - Part 2," *Computers & Industrial Engineering,* vol. 53, pp. 241-246, 2007.

[81] M. Alberti, J. Ciurana, and M. Casadesus, "A system for optimising cutting parameters when planning milling operations in high-speed machining," *Journal of Materials Processing Technology,* vol. 168, pp. 25-35, 2005.

[82] A. Vidal, M. Alberti, J. Ciurana, and M. Casadesus, "A decision support system for optimising the selection of parameters when planning milling operations," *International Journal of Machine Tools and Manufacture,* vol. 45, pp. 201-210, 2005.

[83] P. Tandon and M. Rajik Khan, "Three dimensional modeling and finite element simulation of a generic end mill," *Computer-Aided Design,* vol. 41, pp. 106-114, 2009.

[84] M. Kurt and E. Bagci, "Feedrate optimisation/scheduling on sculptured surface machining: a comprehensive review, applications and future directions," *The International Journal of Advanced Manufacturing Technology,* vol. 55, pp. 1037-1067, 2011.

[85] H.-U. Lee and D.-W. Cho, "Development of a reference cutting force model for rough milling feedrate scheduling using FEM analysis," *International Journal of Machine Tools and Manufacture,* vol. 47, pp. 158-167, 2007.

[86] B. Arezoo, K. Ridgway, and A. M. A. Al-Ahmari, "Selection of cutting tools and conditions of machining operations using an expert system," *Computers in Industry,* vol. 42, pp. 43-58, 2000.

[87] H. Sakurai, "Decomposing a delta volume into maximal convex volumes and sequencing them for machining," in *Proceedings of the 1994 ASME International Computers in Engineering Conference and Exhibition. Part 1 (of 2),* 1994, pp. 135-142.

[88] Z. Huang, B. Xie, L. Ma, and X. Wei, "Feature conversion based on decomposition and combination of swept volumes," *CAD Computer Aided Design,* vol. 38, pp. 857-873, 2006.

[89] A. C. Lin, S.-Y. Lin, D. Diganta, and W. F. Lu, "An integrated approach to determining the sequence of machining operations for prismatic parts with interacting features," *Journal of Materials Processing Technology,* vol. 73, pp. 234-250, 1998.

[90] P. Dave and H. Sakurai, "Maximal volume decomposition and its application to feature recognition," in *Proceedings of the 1995 Database Symposium*, 1995, pp. 553-568.

[91] E. Wang, Y. S. Kim, and Y. Woo, "Feature recognition using combined convex and maximal volume decompositions," in *DETC2005: ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2005, pp. 469-475.

[92] L.-W. Tyan and V. Devarajan, "Automatic identification of non-intersecting machining features from 2D CAD input," *CAD Computer Aided Design*, vol. 30, pp. 357-366, 1998.

[93] G. H. Ma, Y. F. Zhang, and A. Y. C. Nee, "A simulated annealing-based optimization algorithm for process planning," *International Journal of Production Research*, vol. 38, pp. 2671-2687, 2000.

[94] H. C. Wu and T. C. Chang, "Automated setup selection in feature-based process planning," *International Journal of Production Research*, vol. 36, pp. 695-712, 1998.

[95] C.-C. P. Chu and R. Gadh, "Feature-based approach for set-up minimization of process design from product design: Computer-Aided Concurrent Design," *Computer-Aided Design*, vol. 28, pp. 321-332, 1996.

[96] Y. Rong, S. H. Huang, and Z. Hou, *Advanced Computer-Aided Fixture Design*: Academic Press, 2005.

[97] T.-C. Chang, H.-P. Wang, and R. A. Wysk, *Computer-aided manufacturing* 3rd ed., 2006.

[98] P. J. Schneider and D. H. Eberly, *Geometric Tools for Computer Graphics*: Morgan Kaufmann, 2003.

[99] R. Goldman, "Curvature formulas for implicit curves and surfaces," *Computer Aided Geometric Design*, vol. 22, pp. 632-658, 2005.

ANNEX 1 – CONCAVITY OF HOOP EDGES

In section 3.3.8.1 we proved, in the situation that an island extrudes from one face, edges consisting a hoop have the same concavity. Now we are going to prove this conclusion is also true even when the island extrudes from multiple faces.

Denote $\{f_i\}$ the set of faces bounding the island, $\{g_j\}$the set of faces from which the island extrudes. Denote $\{e_k\}$ the set of edges consisting of the hoop of the island. Obviously we have

$$e_k \in (\cup f_i) \cap (\cup g_j)$$

and

$$\cup e_k = (\cup f_i) \cap (\cup g_j)$$

An island is said extruding from $\{g_j\}$if at any point p in any edge $e \in \{e_k\}$, the face $g \in \{g_j\}$ passing p has its normal vector **m** corresponding to the vector $\mathbf{n}_e$ normal to e on the right side and tangent to the face $f \in \{f_i\}$ passing p, see Figure Annex 2. That is

$$\mathbf{m} \bullet \mathbf{n}_e > 0 \qquad \text{(A.1)}$$

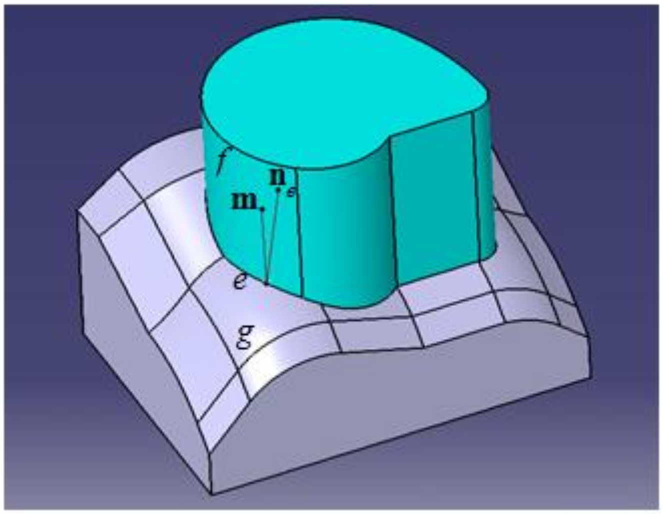


Figure Annex 1 Island extruded from face *g*

There are three situations that faces of $\{f_i\}$ intersect with those of $\{g_j\}$. First, several faces of $\{f_i\}$ intersect one face of $\{g_j\}$, which has been proved in section 3.3.8.1; second, one face of $\{f_i\}$ intersects two or more adjacent faces of $\{g_j\}$; and third, two adjacent faces of $\{f_i\}$ intersect two adjacent faces of $\{g_j\}$ respectively.

In the second situation, suppose $f_i \in \{f_i\}$ intersects with $g_j, g_{j+1} \in \{g_j\}$ at edges e_j and e_{j+1} respectively. Denote $e_j = f_i \cap g_j$ and $e_{j+1} = f_i \cap g_{j+1}$, see Figure Annex 2, suppose $\|e_j\| \neq 0$ and $\|e_{j+1}\| \neq 0$. It is intuitive $e_j, e_{j+1} \in \{e_k\}$.

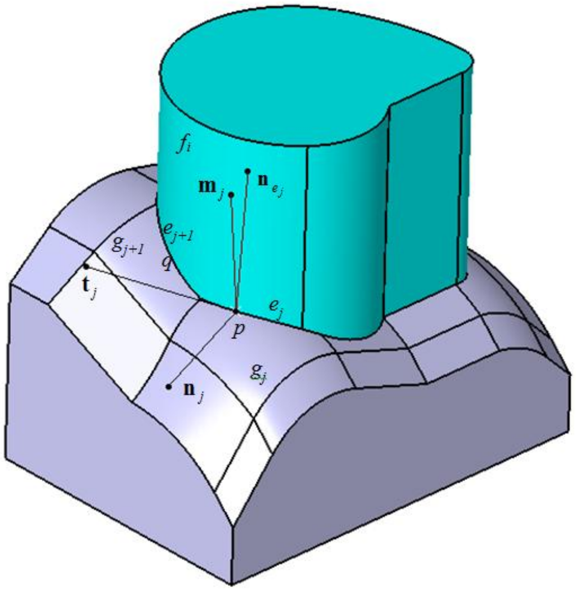


Figure Annex 2 Faces and common point of the second situation

Denote $\mathbf{n}_j$, $\mathbf{m}_j$ the normal vectors of f_i, g_j at a point p in edge e_j respectively, and $\mathbf{n}_{j+1}$, $\mathbf{m}_{j+1}$ the normal vectors of f_i and g_{j+1} at a point q in edge e_{j+1} respectively. Denote $\mathbf{t}_j$, $\mathbf{t}_{j+1}$ the tangent vectors of e_j and e_{j+1} at p and q respectively. As e_j, $e_{j+1} \in \partial(f_i)$, thus the interior of f_i situates on the right side of $\mathbf{t}_j$ and $\mathbf{t}_{j+1}$. As it is assumed the island extrudes from $\{g_j\}$, apply (A.1) to edge e_j, face f_i and g_j, we have

$$\mathbf{m}_j \mathbf{n}_{e_j} > 0 \qquad \text{(A.2)}$$

Where $\mathbf{n}_{e_j}$ is the vector, passing *p*, tangent to f_i and normal to e_j, thus $\mathbf{n}_{e_j} = \mathbf{t}_j \times \mathbf{n}_j$. Substitute $\mathbf{n}_{e_j}$ in (A.2) by $\mathbf{t}_j \times \mathbf{n}_j$, we get

$$\mathbf{m}_j(\mathbf{t}_j \times \mathbf{n}_j) > 0$$

Similarly we have

$$\mathbf{m}_{j+1}(\mathbf{t}_{j+1} \times \mathbf{n}_{j+1}) > 0$$

Rewrite these two inequalities as

$$(\mathbf{n}_j \times \mathbf{m}_j)\mathbf{t}_j > 0 \tag{A.3}$$

And

$$(\mathbf{n}_j \times \mathbf{m}_{j+1})\mathbf{t}_{j+1} > 0 \tag{A.4}$$

To prove edges e_j and e_{j+1} have the same concavity, we assume their concavities are different. Not lose generality suppose e_j is a concave edge and e_{j+1} is convex. Apply inequalities (3.15) and (3.17) to these two edges and corresponding faces, we have

$$(\mathbf{n}_j \times \mathbf{m}_j)\mathbf{t}_j > 0 \tag{A.5}$$

And

$$(\mathbf{n}_{j+1} \times \mathbf{m}_{j+1})\mathbf{t}_{j+1} < 0 \tag{A.6}$$

The contradiction between (A.6) and (A.4) implies that the assumption of e_{j+1} being convex edge is not true. So both e_j and e_{j+1} must be concave edges.

In the third situation, give f_i, $f_{i+1} \in \{f_i\}$, $g_j, g_{j+1} \in \{g_j\}$, their adjacencies are determined by

$$\begin{aligned} e_f &= f_i \cap f_{i+1} \\ e_g &= g_j \cap g_{j+1} \\ e_j &= f_i \cap g_j \\ e_{j+1} &= f_{i+1} \cap g_{j+1} \end{aligned}$$

Denote $\mathbf{n}_j$, $\mathbf{m}_j$ the normal vectors of f_j, g_j at an arbitrary point p in edge e_j respectively, and $\mathbf{n}_{j+1}$, $\mathbf{m}_{j+1}$ the normal vectors of f_{j+1} and g_{j+1} at an arbitrary point q in edge e_{j+1} respectively. Denote $\mathbf{t}_j$, $\mathbf{t}_{j+1}$ the tangent vectors of e_j and e_{j+1} at p and q respectively. Vectors

$\mathbf{n}_{e_j}$ and $\mathbf{n}_{e_{j+1}}$, normal to e_j and e_{j+1} and tangent to f_i and f_{i+1} respectively, can be determined by $\mathbf{n}_{e_j} = \mathbf{t}_j \times \mathbf{n}_j$, $\mathbf{n}_{e_{j+1}} = \mathbf{t}_{j+1} \times \mathbf{n}_{j+1}$. According to the criterion (A.1), at every point in edges e_j and e_{j+1}, we have

$$\begin{cases} \mathbf{m}_j(\mathbf{t}_j \times \mathbf{n}_j) > 0 \\ \mathbf{m}_{j+1}(\mathbf{t}_{j+1} \times \mathbf{n}_{j+1}) > 0 \end{cases} \tag{A.7}$$

Which are equivalent to

$$\begin{cases} (\mathbf{n}_j \times \mathbf{m}_j)\mathbf{t}_j > 0 \\ (\mathbf{n}_{j+1} \times \mathbf{m}_{j+1})\mathbf{t}_{j+1} > 0 \end{cases} \tag{A.8}$$

The expressions in (A.8) confirm that both e_j and e_{j+1} are concave edges.

ANNEX 2 – ALGORITHM 3.10

Algorithm 3.10 – Determine the inlaid face EF of an island

```
    Given a hoop H
//A hoop consists of a list of edges
    get edge E[0] of H
    get vertex V[0] of E[0]
    get edge e0 passing V[0], $e0 \notin H$
    get face f0 bounded by E[0] and e0
//f0 is one of the column faces of the island
    get vertex v0[0] of e0 as referenced in f0, $v0[0] \neq V[0]$
//orientation of e0 is referenced by f0: the interior of f0 is on the right side of e0 as //viewed from outside of current DV
//e0 is preceding E[0] in the loop bounding the f0
    make a unit vector r tangent to e0 at v0[1]
    make a plane p passing v0[1] and perpendicular to r
//v0[ ] is the VertexList of e0
    get edge e1 preceding e0 in the loop bounding f0
//e1 passes v0[0], $e1.Vlist = \left(v1[0],\ v1[1]\right)$, v1[1] = v0[0], and $e1 \in Loop(f0)$
    Make edge E1 = $\left(v1[1],\ v1[0]\right)$
//E1 belongs to the loop of face f1 which is the end face of the island
    $\left\{ f_i \right\} = getface(v0[0])$
    For each $f_i$
        $l_i$ = getloop($f_i$)
        if E1 $\in$ $l_i$
```

```
            F1 = $f_i$
//F1 is the or one of the end faces of the island
            make an edge chain list ECList
//edges of ECList bound the end faces of the island
    End for
    ECList[1] = E1
    Orientate E1 in such a way that v0[0] of e0 is its from vertex
//v1[0] = v0[0]
//the following loop is developed for the purpose of getting the closed edge chain denoted
//ECList, which is the rim of the island, at the other end of the island where e1 exists
    For I = 1, …, N
        determine edge ECList[i+1] in such a way that:
      1)  its "from" vertex =  "to" vertex of ECList[i]
      2)  not in the loops enclosing the left and right faces of ECList[i]
         If ECList[i+1] shares v1[0] with ECList[1]
//the chain is closed
            End for
        Else
            Continue
        End if
   End for
   prjCL = project(ECList, p, $\mathbf{r}$)
   prjH = project(H, p, $\mathbf{r}$)
   prjSideF = project(sideFace of island, p, $\mathbf{r}$)
//function project($x$, $\boldsymbol{y}$, $\mathbf{r}$) projects $x$ onto $y$ in direction $\mathbf{r}$
```

```
     Get the extremum of H in the direction of r: R = extreme(H, r)
     Make plane q passing R and perpendicular to r
    If e0 is a linear segment and if prjCL = prjSideF
//the island is a straight column-wise volume
          Extend side face(s) of the island to q to make new side face(s)
          Make face EF by trimming q with the surfaces of side faces
     Else
//the island is either a part of a cone, a pyramid or a volume with side face(s) of two
//dimensional curvature
          Arrange side faces f[i] in a list Flist in accordance with their adjacency
          Get surface of each f[i] and put them in SFList
          If the island is a cone
            Find its axis AX and determine a unit vector r on AX
// r points toward H from the vertex of A
            Create a plane q passing R = extreme(H, r) and perpendicular to AX
            Extend the cone to q to make new side face for the island
          Else
//island is not a cone
               Make plane PL passing v0[1], the centers of F1 and prjH
               Get intersection of PL with the side faces of the island: c1 and c2
//generally c1 and c2 are curves, and c1 = e0
               Get the intersections of c1 and c2 with H: p1 and p2 respectively
            Make an angle A in PL with two sides that are tangent to c1 and c2 at p1 and
            p2 respectively
               Determine a unit vector r on the bisector of A
```

```
// r points toward H from the vertex of A
        End if
        If the vertex of A situates between the faces enclosed by H and ECList
//the vertex is in the interior of the island
            Get the extremum of H in the direction of r: R = extreme(H, r)
        Else
            Get the extremum of H in the direction of r: R = extreme(H, -r)
        End if
        Make plane q passing R and perpendicular to r
        Make the inlaid face EF of the island by trimming q with Slist
//Slist is the set of surfaces of side faces of the island
    End if (e0 is linear)
End of algorithm 3.10
```

ANNEX 3 – ALGORITHM 3.12

Algorithm 3.12 – Determination of splitting face of a co-surfaced edge chain

```
    Given face F
    Given outer loop L of F
   Given a concave vertex CV in loop L
    Get concave edge chain Echain starting at CV
    If CV is a front concave vertex
        FCE = getface.left(E[1])
//edge E[1] is the first edge of Echain, which consists of CV as the “from” vertex
        SideIndicator = Right
//the side of an edge chain is determined correspond to the chain direction
    Else
        FCE = getface.right(E[1])
        SideIndicator = Left
    End if
    S = getsurface(FCE)
//Get the surface S of FCE
    M = 0
    If CV is a front concave vertex
        For each edge E[i] of Echain, I = 2, …, n
            Get the left face LF[i] of E[i]
            If $S[1] \cap LF[i] - LF[i] \neq \emptyset$
                M = I -1
//the amount of M is the number edges that do not belong to the selected surface S
```

```
                Else
                End if
            End for
        Else
//CV is a back concave vertex
            For each edge E[i] of Echain, I = 2, …, n
                Get the right face RF[i]
                If $S[1]\cap RF[i]-RF[i]\neq\emptyset$
                    M = I -1
                Else
                End if
            End for
        End if
        If M = 0
//edges of Echain are co-surfaced within S
//get the portion of S bounded by block faces, denoted SB
            SB = $S\cap B$
//get the intersection of SB and current DV
            InterS = intersect(SB, DV)
//then get the complement of F in InterS
            CompF = InterS $-^{*}$ FEC
//usually, CompF is consisted of several faces
//in CompF the portion enclosed by Echain is then the splitting face
          LP = getloop(E[1], CompF)
        spltF = Getface(LP)
    End if
END of algorithm 3.12
```

ANNEX 4 – ALGORITHM 3.13

Algorithm 3.13 – Determine face pieces of an Echain

```
    Given an Echain
//e[1] and e[N] are the first and the last edges in Echain
//Echain is oriented from e[1] to e[N]
    Get the number N of edges in Echain
    For each edge e[i] in Echain, $i \le \left\lfloor \frac{N}{2} \right\rfloor$
//$\left\lfloor \frac{N}{2} \right\rfloor$ is the integral part of $\frac{N}{2}$
        Join the “from” vertex of e[i] and the “to” vertex of e[N-i+1] with a line segment J[i]
        If J[i], e[i] and e[N-i+1] are co-surfaced on surface S
            Find out the largest number of co-surfaced edges connected one with the other in Echain
            If all these edges make up a string C consisted of e[i] and e[N-i+1]
                Make a face piece bounded by C and J[i]
            Else
//There are two separated edge strings C1 and C2, C1 is attached with e[i], C2 with  e[N-i+1]
                Join the “from” vertex of C1 and the “to” vertex of C2 with a line segment J[i]
                Join the “to” vertex of C1 and the “from” vertex of C2 with a line segment J[i+1]
                Make a face piece bounded by C1, C2, J[i] and J[i+1]
            End if
        Else
```

```
                Make a face piece by bounding the LS[i] of e[i] with LS[i-1], LS[i+1], RS[i] and RS[N-i+1]
            End if
            If N is an odd number
                If e[$\left\lfloor \frac{N}{2} \right\rfloor$+1] and e[$\left\lfloor \frac{N}{2} \right\rfloor$] are not co-surfaced
                    Get intersection of LS[$\left\lfloor \frac{N}{2} \right\rfloor$+1] and LS[$\left\lfloor \frac{N}{2} \right\rfloor$], denoted intE
                    Make face piece bounded by intE and e[$\left\lfloor \frac{N}{2} \right\rfloor$+1]
                End if
            End if
        End for
END
```

ANNEX 5 – ALGORITHM 4.2

Algorithm 4.2 Judgment of bottom narrowed SDV

```
Given a SDV, together with the set of its open faces SDV.open_face,
For each open face $f_j \in$ SDV.open_face
//each open face is considered as an access face
      f = $f_j$
    p = getgeocenter(f)
    **n** = normal of f at p
    Q = make_plane(p, **n**)
    f_project_n = project(f, Q, **n**)
    B = getbottomface(SDV, f)
    bottom_project_n  = project(B, Q, **n**)
    If the SDV is linear sided{
            If bottom_project_n $\subseteq$ f_project_n
                bottom_narrowed = "true"
                Tool_axis vector **s** = **n**
            else
              p = getgeocenter(B)
              **m** = normal of B at p
              Q = make_plane(p, **m**)
              f_project_m  = project(f, Q, **m**)
              bottom_project_m  = project(B, Q, **m**)
              If bottom_project_m $\subseteq$ f_project_m
                  bottom_narrowed = "true"
```

```
                    Tool_axis vector **s** = **m**
                Else
                    sum_projection_n  = $\bigcup$projection(SDV.face\f, Q, **n**)
//SDV.face = all boundary faces of SDV except face f
                    If sum_projection_n $\subseteq$ f_project_n
//sum_projection_n  – f_project_n = $\varnothing$
                        bottom_narrowed = “true”
                        Tool_axis vector **s** = **n**
                    Else
//the projection of B onto Q in direction **n** may be shift to one side
                            sum_projection_m  = $\bigcup$projection(SDV.face\B, Q, **m**)
                        If sum_projection_m $\subseteq$ bottom_project_m
//sum_projection_m  – bottom_project_m = $\varnothing$
                            bottom_narrowed = “true”
                            Tool_axis vector **s** = **m**
                             else
                                 SDV is not bottom narrowed
    end IF-Else
    If bottom_narrowed = “true”
        SDV[i].faceindex_bottnarrow = *j*
    Else
         SDV[i].faceindex_bottnarrow = 0
        Tool_axis vector s = 0
}
```

```
End for

END of algorithm 4.2
```