3-Dimensional surface simulation

TUE Internal Report

3-Dimensional Surface Simulation

TUE-Research Report

Gaohong Xie [1]

Nov. 1991

WPA 91-1188

WPA, Dept. of Mechanical Engineering
Eindhoven University of Technology
P.O.Box 513, 5600 MB, Eindhoven
The Netherlands

[1] the author is on leave from Dept. of Mechanical Engineering, Tangshan Institute of Technology, 063009 Tangshan, P.R.C. He is now with Department of Mechanical Engineering in Eindhoven University of Technology , e-mail gaohong@wfw.wtb.tue.nl

3-Dimensional Surface Simulation

Gaohong Xie M.Sc.

Abstract

The construction and simulation of a solid surface is the issue that arise in numerous applications. This report deals with three basic surfaces of structures, Plane, cylindrical surface and spherical surface. To simulate the surfaces, their data structure are discussed, a general approach to define some parameters, such as outward normal vector, tagential vectors and rate of change of normal vector to tangential vectors are developed and described in detail.

Contents

List of Figures

1 Introduction

The requirement for geometric modelling of solid surfaces arises in numerous areas such as graphical image, surface interpolation and computer aided design [1]. In engineering, the frequently used finite element method in 3-dimensional problems needs definitions on the surfaces and other parameters for the descriptions of constraints, friction and the calculation of tangent stiffness matrix. For example, in the finite element analysis of deep drawing process with software package ABAQUS, the user has to define rigid surfaces by an user interface subroutine RSURFU, in which the following definitions have to be supplied to ABAQUS:

1. The coordinates of closest point on the rigid surfaces (penetration point) to the given deformed point.

2. The Euclidean distance between the deformed point and penetration point on rigid surfaces.

3. The local surface geometry defintions of rigid surface at penetration point, which include normal vector, tangential vectors and rate of normal vector to tangential vectors.

In most cases, programming such an interface subroutine for a complex geometry is labour cost and error pron, therefore, a general approach is described based on an assumption that any 3-dimensional surface can be classified as union sets of planes, spherical or cylindrical surface parts.

The usual starting point for defining a solid model is provided by a description of the surface as a series of slices or sections. To make algorithm simple and efficient, we describe the basic surface as a single section instead of some kind of patch without losing any generality. The method is valid as long as the considered surface is one or combination of three basic surface sections: planar bounded planes, bounded partial cylindrical side surfaces and partial spherical surfaces. For planes and cylinders, they can be easily described in local coordinate systems, however, local defined orthogonal vectors will be rotation dependent when mapping procedure from local coordinate system to global coordinate system is employed to transfer local definitions into global definitions, therefore, in the following sections, the descriptions are refered to global frame.

2 Bounded Planes

In engineering structures, most surfaces can be expressed by bounded 3-D planar planes. The bounded plane usually can be viewed as a simple polygon via its node vertices as in Fig 1. The data structure for bounded plane, therefore, is an ordered set of points $\{p_i\}_{i=1\dots n}$. The outward normal from view of specified point can be specified by the order of points $\{p_i\}$. It is noticed that a bounded plane at least has three vertice points (triangle).

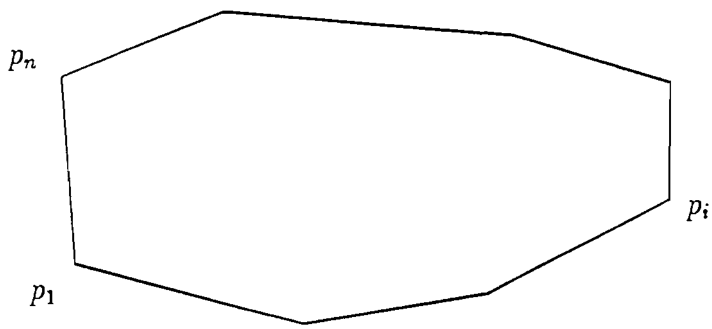

Figure 1: Representation of a bounded plane, simple polygon

For each given point p we want to determine a point p_j, the penetration point on bounded plane (see Fig. 2), which satisfy:

$$d(p, p_i) \leq d(p_j, p) \tag{1}$$

Where $d(X, Y)$ is the Euclidean distance between the two points X and Y. To find projection point p_j of a given point p on the plane, we choose three vertices of polygon p_{i-1}, p_i, p_{i+1} which are not in a common line as shown in Fig. 3. The following equations results three linear equations about coordinates of projection point p_j

$$d(p_l, p)^2 - d(p_l, p_j)^2 = h^2, \quad l = i-1, i, i+1 \tag{2}$$

For bounded planes, we need to decide that this projection point p_j is inside bounded plane (polygon) or outside polygon. The problem is termed as point inclusion in computational geometry. Several algorithms are available as described in [2]. The mostly used methods are intersection method and slab method. In intersection method, the outside polygon point p_o is choosen, then find intersection numbers between line segment $p_o\ p_j$ and polygon (see Fig. 4). If the times of intersection are odd numbers, the point p_j is inside polygon else it is outside polygon.

In slab method, the bounded plane is divided into triangles (star shaped polygon), therefore, point inclusion problem is transformed to that checking a given point is inside a triangle or outside it as shown in Fig. 5.

If the projection point is outside the polygon, then other sections are considered. The plane's outward normal $\vec{N}$ is determined either by the vector from given point p to projection point p_j or by two ordered polygon side vectors since they are ordered in certain

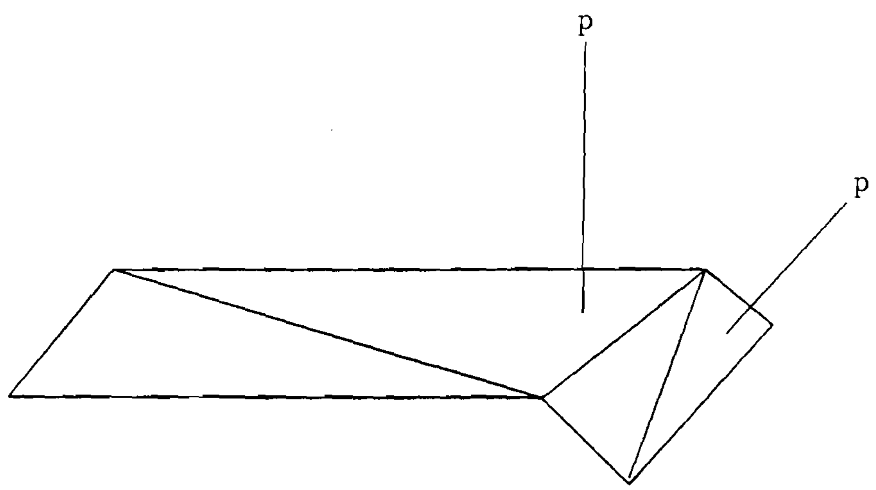

Figure 2: Projection of points on bounded planes

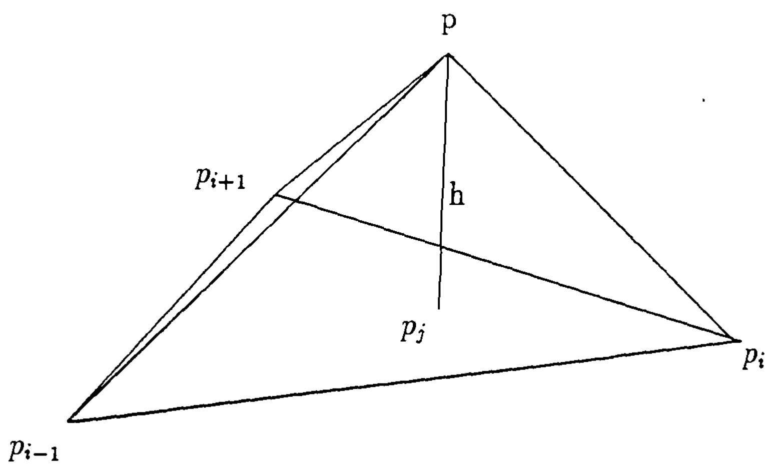

Figure 3: Find projection p_j of a given point p

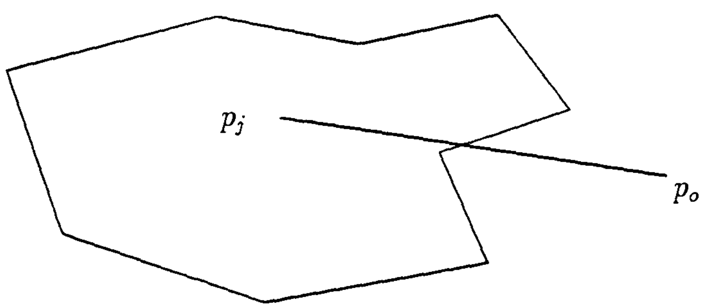

Figure 4: Point inclusion problem

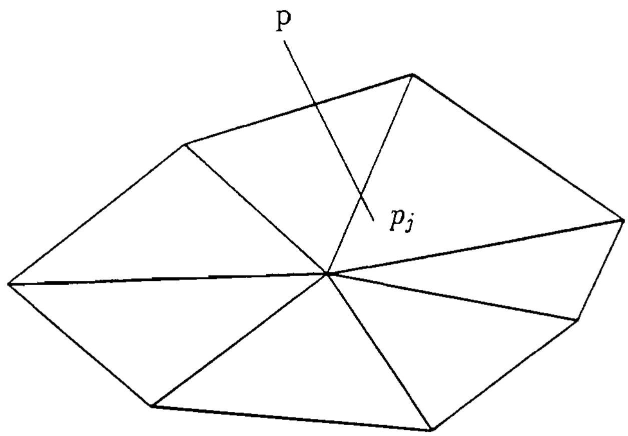

Figure 5: Slab method for Point inclusion problem

direction.

While the projection point is in the bounded plane, then the first tangential vector $\vec{T_1}$ is set with components $(x, y, 0)$ while second tangential vector is decided by right hand rule of vector cross product as $\vec{T_2} = \vec{N} * \vec{T_1}$. Components x, y of vector $\vec{T_1}$ can be solved from the following equations:

$$\vec{N}.\vec{T_1} = 0 \tag{3}$$

$$x^2 + y^2 = 1 \tag{4}$$

The rate of normal vector to tagential vectors is calculated via:

$$\frac{\partial \vec{N}}{\partial \vec{T}} = (\partial N_x/\partial \vec{T}, \partial N_y/\partial \vec{T}, \partial N_z/\partial \vec{T}) \tag{5}$$

Here N_x, N_y, N_z are components of vector $\vec{N}$ on global axises, and $\partial N_i/\partial \vec{T} = \partial N_i/\partial x_i \cos\alpha_i$ with $x_i = x, y, z$ and $\cos\alpha_i$ direction cosine of vector $\vec{T}$.

3 Bounded Cylindrical Surface

Bounded cylindrical side surface (with two ends surfaces vertical to the axis) is expressed by a pair sets of three points $(p_1, p_2, p_3), (p_4, p_5, p_6)$, with p_2, p_5 middle points of arc (p_1, p_2, p_3) and (p_4, p_5, p_6) as in Fig. 6.

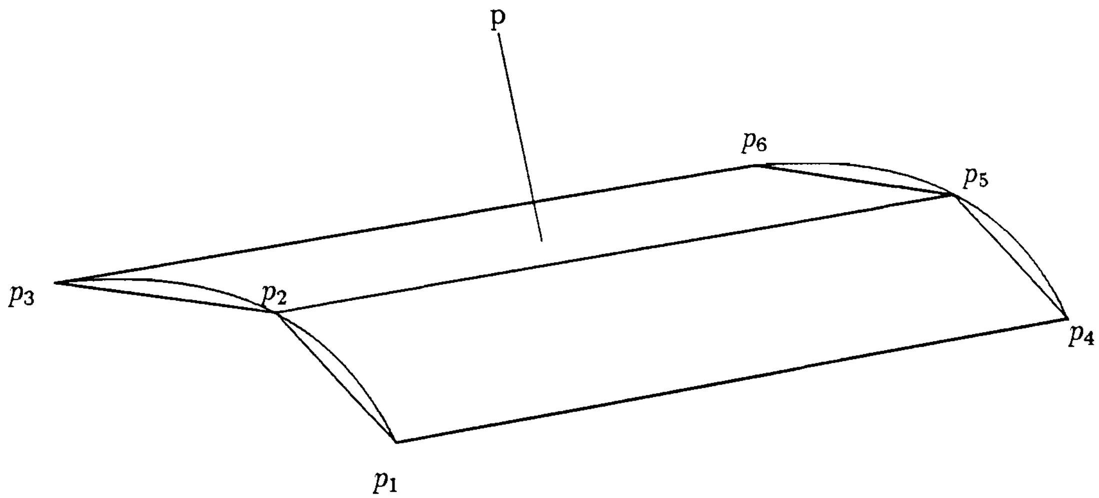

Figure 6: Cylindrical side surface

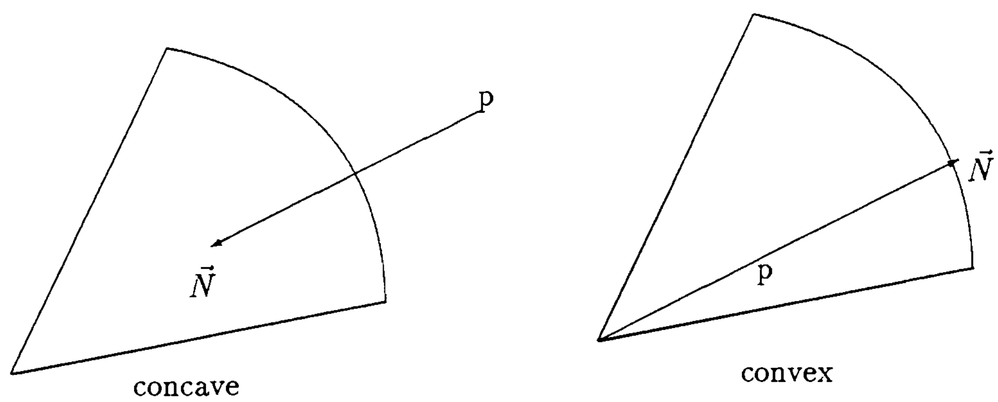

Figure 7: Shape of cylindrical surface from deformed material point view

Although the radius of cylinder can be calculated from the pair sets of points' coordinates, the radius is normally known as input. The cylinderical axis can be expressed by two center points of the arcs. To define its outward normal from the view of deformed material point which will contact with cylindrical surface, another parameter which describe the shape of cylindrical surface (convex and concave) is needed as in Fig. 7.

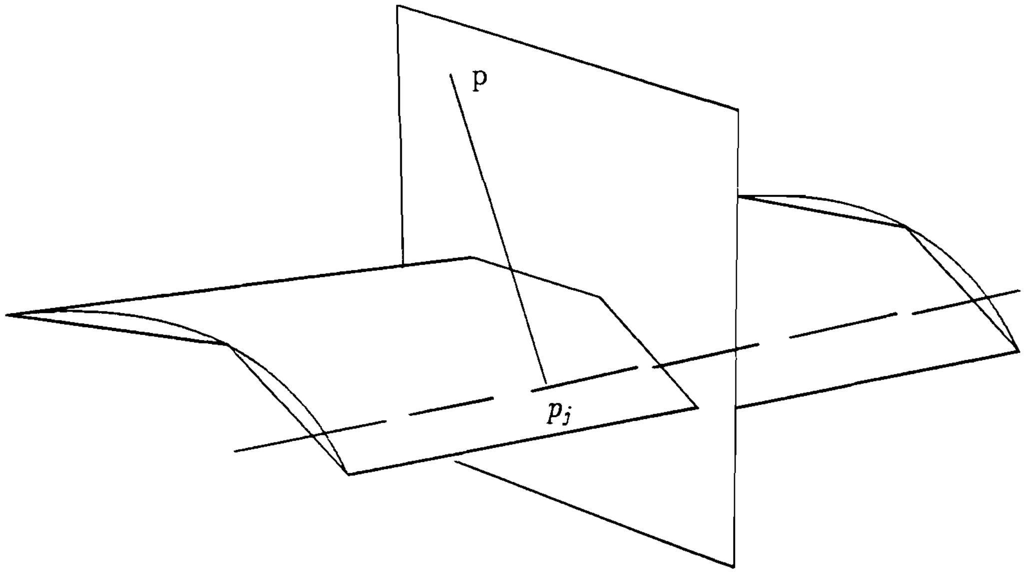

Figure 8: Intersection of cylindrical axis and the plane

To decide whether the given point has a projection point on cylindrical surface, the bounded planes through points (p_1, p_2, p_4, p_5) and (p_2, p_3, p_5, p_6) are used to calculate the given point's projection point on the two planes (see also Fig. 6). Since the one to one mapping between the two bounded planes and bounded cylindrical side surface. The problem of whether the given point has a projection point on cylindrical side surface is transformed to whether a given point has a projection point on two bounded planes. The cylindrical surface's axis can calculated via the data of cylindrical surface, the center coordinates of arc on the plane which vertical to the cylindrical's axis and pass the given point is found by finding the intersection point of the axis and the plane as in Fig. 8.

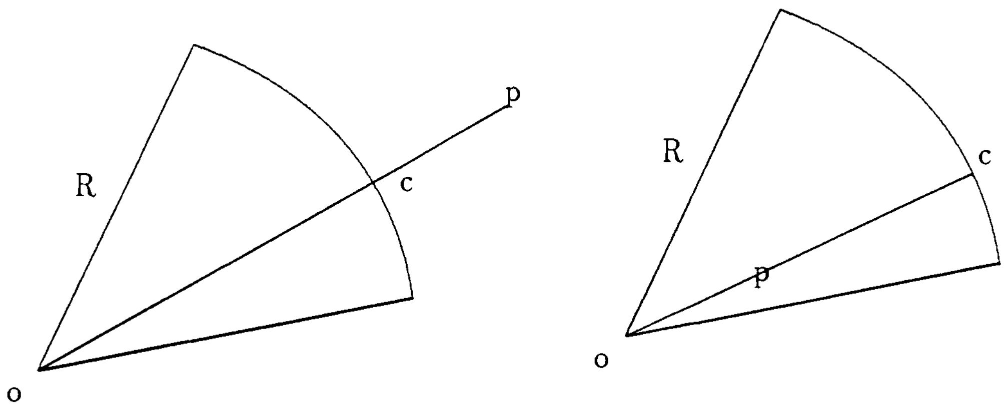

Figure 9: Projection point of cylindrical surface

The projection point c on the cylindrical side surface of a given point p can be calculated by finding the point c on cylindrical surface, which divide the vector $\vec{op}$ to a ratio $R/d(o,p)$. Here R is the radius of cylindrical surface and $d(o,p)$ is the distance between points o and p. (see Fig. 9)

The outward normal vector $\vec{N}$ then can be easily defined by vector $\vec{po}$ or $\vec{op}$ according to the shape of cylindrical surface. As in plane's case the first tangential vectors $\vec{T_1}$ is set as $(x, y, 0)$, and use equations(3) and (4) to decide $\vec{T_1}$ while the second tangential vector is decided by right hand rule of vector cross product as $\vec{T_2} = \vec{N} * \vec{T_1}$. Other parameters followed the same methods as described in plane's case.

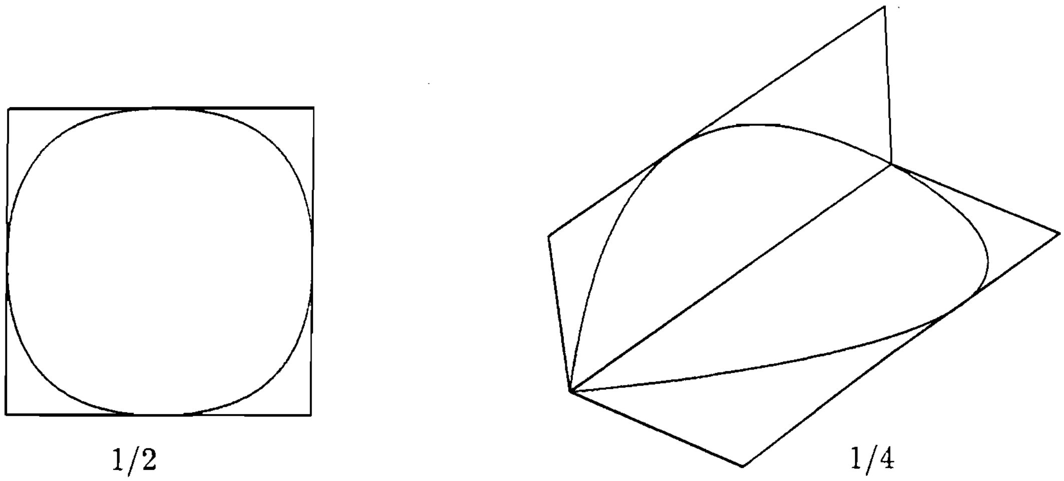

Figure 10: Bounded inclusion planes for 1/2 and 1/4 part of spherical surface

4 Spherical Surface Representation

Any bounded spherical surface can be expressed by its center position, its radius and the portion value. In actual structures, usually 1/2, 1/4, 1/8 parts of spherical surface is used to connect structure surfaces smoothly. For different portions, the algorithm first construct the bounded planes which include the spherical parts (see Fig. 10, 11).

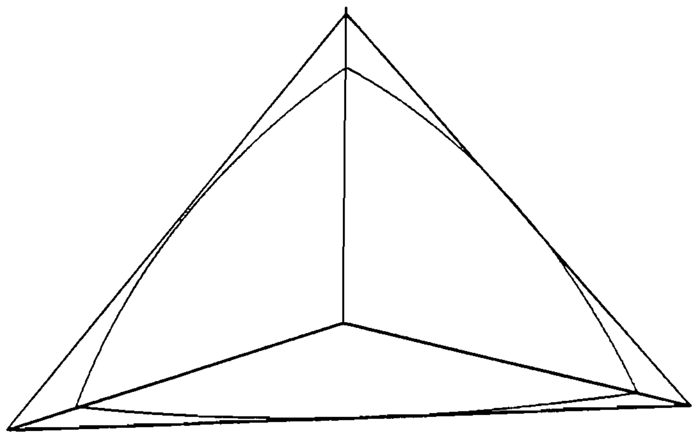

Figure 11: Bounded inclusion plane for 1/8 part of spherical surface

The necessary condition for a point to have a projection point on the spherical parts is this point has projection on the bounded planes. If the necessary condition satisfied, then the calculated projection point have to be checked to see that it is inside the portion part or not. If it is inside the portion part, the parameters can be calculated in the same way as in plane's case. If it is outside spherical portion part, then its projection point can be found in other surfaces.

5 Simulation of Rigid Surface in Plate Deep Drawing

In plate deep drawing process, the rigid surfaces are the contact surfaces of houlder as in Fig. 12, die as in Fig. 13 and punch as in Fig. 14. For this given configuration of tool geometry, the user needs to define each objects' contact surfaces and rewrite the subroutines for specifying the oncoming contact object. The algorithms for the simulation of rigid surface in deep drawing tool geometry are shown in Fig. 15 and Fig. 16.

6 Conclusions

The surfaces normally used in engineering structures can be classified as a union set of different elements namely bounded planes, bounded cylindrical side surfaces and part of spherical surfaces. The data structure for representation these surfaces are discussed, the general approach to define some parameters on these surfaces are developed. The methods described here can be easily implemented into existing codes, such as ABAQUS.

Acknowledgments

The author wishes to thank Dr. Ir. J. A. H. Ramaekers for his finacial support and Ir. M. kessels for his advises on deep drawing geometry configurations.

References

[1] T. J. Baker, 'Shape reconstruction and volume meshing for complex solids', Int. j. numer. methods eng., 32, 665-675(1991).

[2] F. P. Preparata, M. I. Shamos, 'Computational Geometry', Springer-Verlag New York Inc. 1985, ISBN 0-387-96131-3

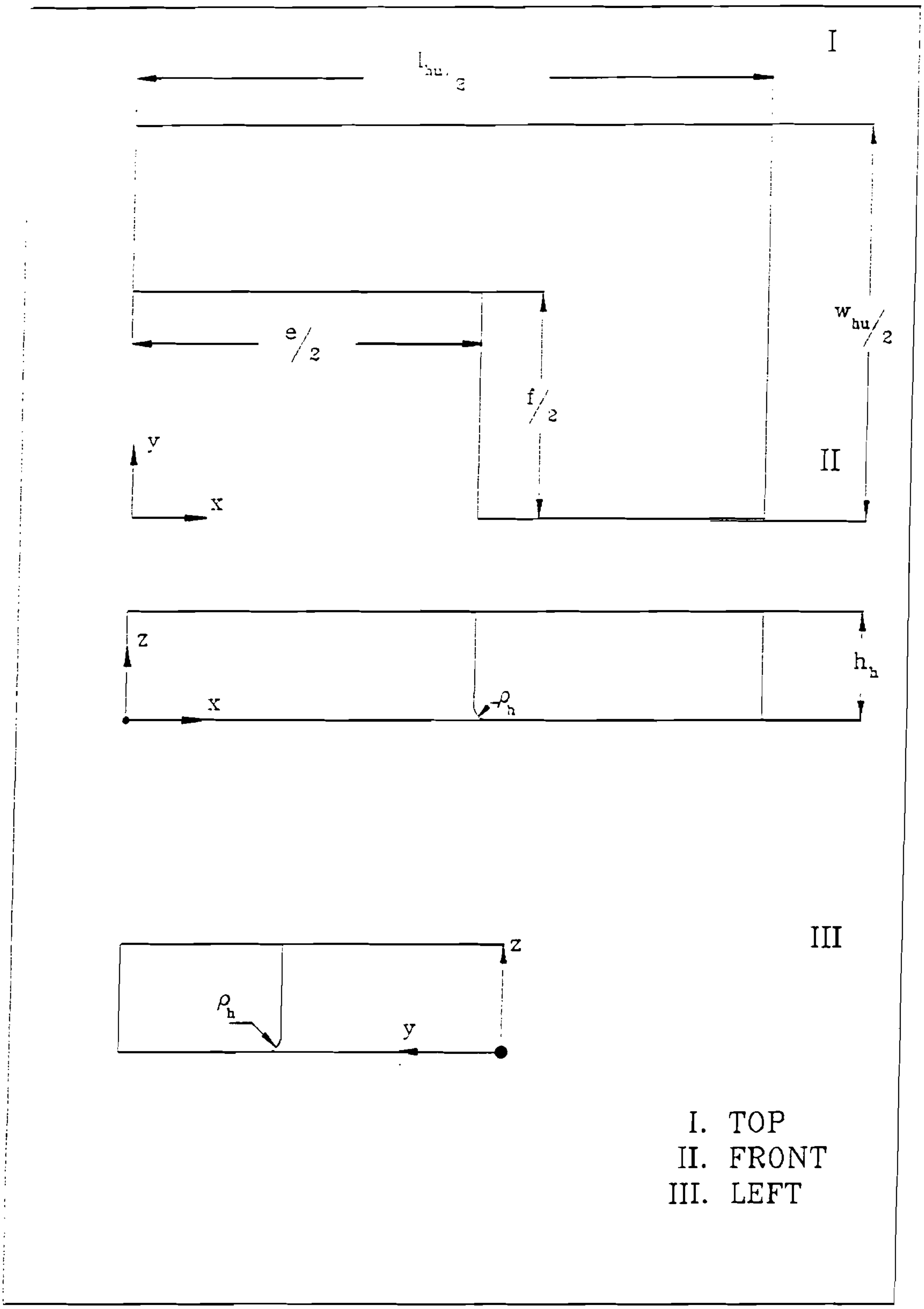

Figure 12: Geometry of houlder

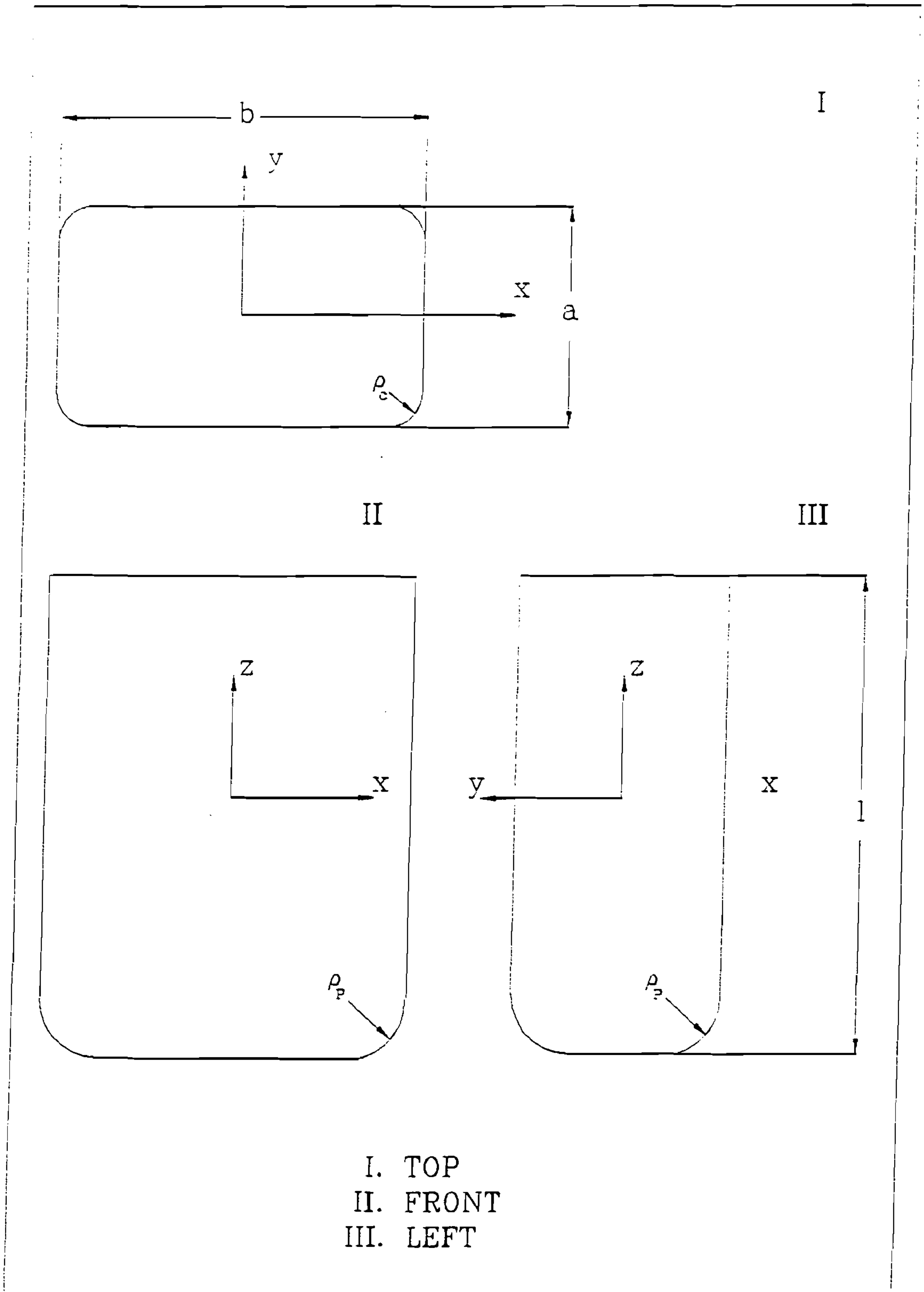

Figure 13: Geometry of die

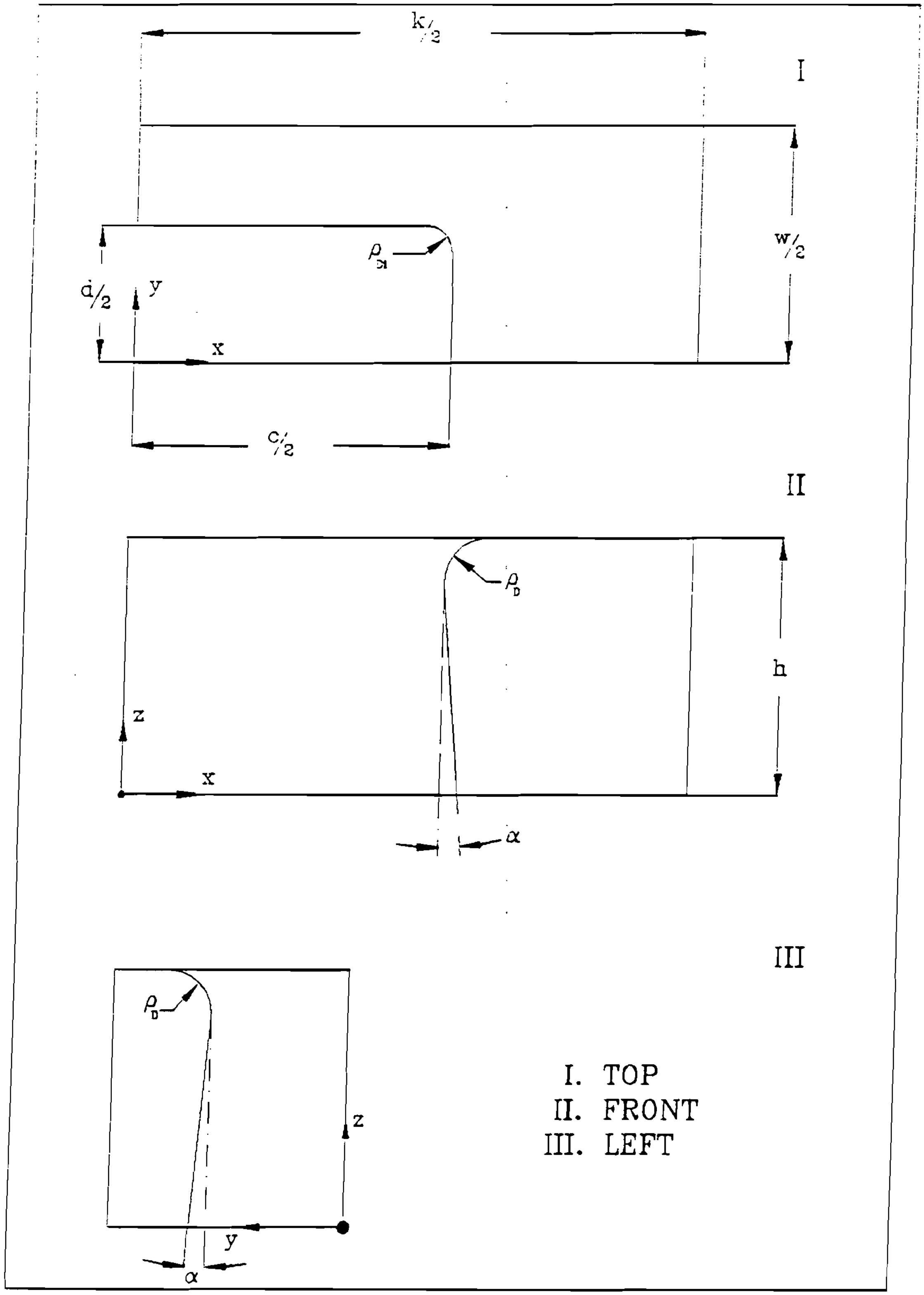

Figure 14: Geometry of punch

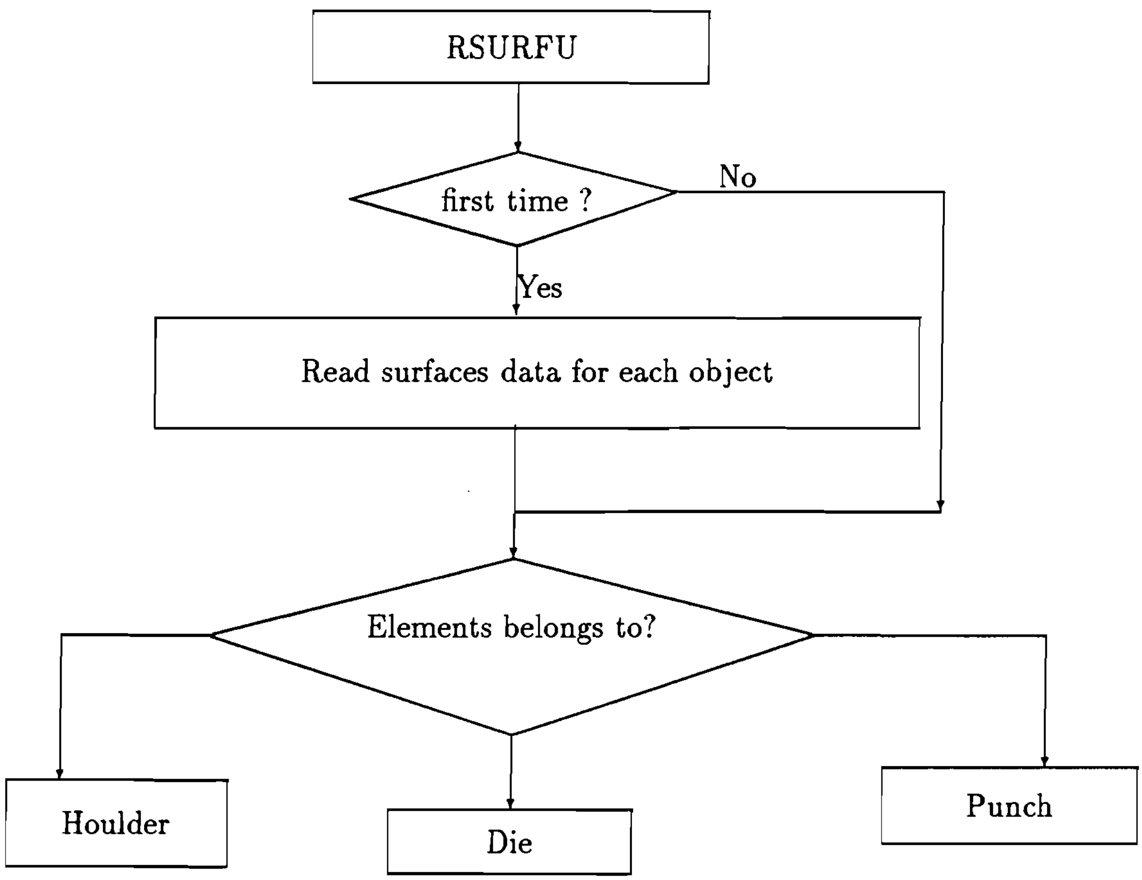

Figure 15: Flow of algorithm for plate deep drawing process

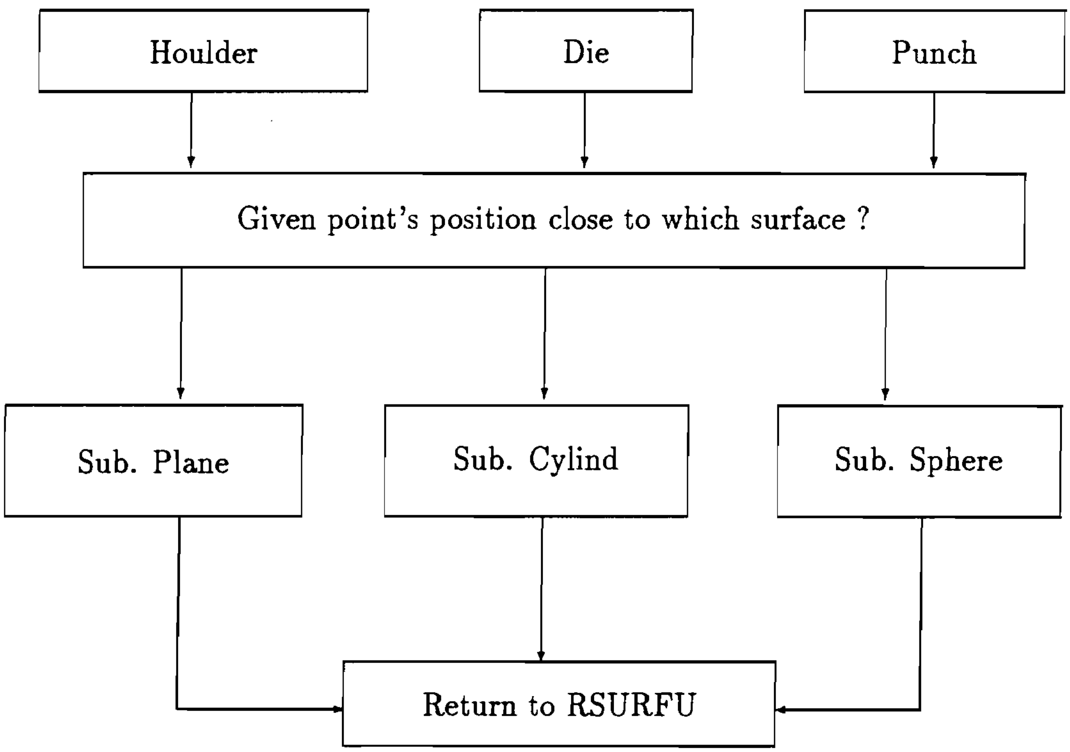

Figure 16: Flow of algorithm for plate deep drawing process