1-1-1993

Development of manufacturability constraints for press forming of sheet metal components

Nirmal Kumar Nair
Iowa State University

Development of manufacturability constraints for press forming of sheet metal components

by

Nirmal Kumar Nair

A Thesis Submitted to the

Graduate Faculty in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Department: Mechanical Engineering

Major: Mechanical Engineering

Iowa State University

Ames, Iowa

1993

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

In many industries, the design, analysis, development and production of sheet metal surfaces comprise a substantial portion of component manufacture. The most prevalent manufacturing process for sheet metal components is press forming. In spite of its widespread use, press forming practice has remained somewhat of an art, typically handled by experienced tooling engineers and designers. This is due to the fact that the physical process of press forming is not well understood. Complex interacting mechanisms and features such as friction, metal flow, material properties and boundary conditions make the press forming process difficult to analyze and predict.

In press forming, the initial flat sheet of material used to develop the final shape is called the blank. The main components of the press forming assembly are the punch, die, and the draw binder mechanism which controls the flow of the blank material inward to form the product. Design of this forming assembly is directly dependent on the surface definition of the final product. For a surface design to be manufactured without defects, the blank should be uniformly deformed by the descending punch without thinning or wrinkling. This process is influenced, to a large extent by the draw binder ring, which is placed outside the trim line, i.e., the boundary between the formed surface and surrounding scrap material. To locate the trim line, the designer must determine the boundary of the area on the blank which is affected by the forming process. This process is referred to as blank de-

velopment. Besides binder-wrap design, the developed blank is also used for punch contact analysis, press forming layout, and as an indicator of material flow during the process.

Iterative redesign of a product which fails in production is very expensive in terms of time and capital investment. It would therefore, be highly useful for product designers to quickly determine the formability of the surface in the early stages of the surface design process. Detailed analysis techniques such as the finite element method are computationally demanding and not amenable for an interactive design environment. Thus, there is a need for quick and qualitative tools which can guide a designer toward a successful design. This research presents a technique to bridge the gap between final design analysis and initial surface construction.

This thesis presents methods to predict the manufacturability of the surface based solely on the geometry of the intended design. The first part of the thesis addresses the development of a blank shape for the design surface. An efficient algorithm is presented to generate blanks for arbitrary non-developable surfaces. In the second half of the thesis geometric conditions of the surface which affect its manufacturability are derived. A methodology for determining surface strains is presented and a visibility criterion algorithm is applied to form a manufacturability evaluator for the die-set parting surface. A press forming simulation software is described for visualizing the forming process and several examples are presented.

Algorithm implementations were was written in C on Silicon Graphics workstations using the Graphics Library (GL) for the graphical interface. FORMS, a public domain toolkit was employed for the user-interface.

CHAPTER 2
RELATED RESEARCH

This chapter surveys the recent advances in the field of sheet metal blank development and numerical and computer modeling of sheet metal deformation processes.

2.1 Blank Development

Sculptured surface models are employed in a wide variety of applications in the automotive, aerospace and appliance industries. Such surfaces can be broadly classified as developable or non-developable. A developable surface is characterized by the ability to form the shape by bending a plane without creasing or tearing, i.e., the surface can be generated by sweeping straight lines or generators along a curve in space. Mathematically, a surface is developable if its Gaussian curvature (the product of the principal normal curvatures) is zero everywhere (Mortenson, 1985). This property is often exploited in algorithms to map a developable surface onto a plane (Redont, 1989). Several methods for the transformation of developable surfaces have been formulated (Clements, 1981; Clements and Leon, 1987; Chu et al., 1985). For example, Clements and Leon (1987) developed an algorithm based on the relationship between the generating and geodesic lines on the surface to get an accurate blank transformation.

Non-developable surfaces encompass the family of surfaces that have non-zero Gaussian curvatures, and thus cannot be generated by simple bending of a plane without

distortion. However, this does not preclude the use of these surfaces in manufacturing since they have advantages over developable surfaces in terms of styling, aerodynamics and other functional aspects of design. To investigate blank shape, Chu. et al., (1985) formulate a simplistic constant area transformation approach for the mapping of a non-developable surface onto a plane. The method is unique from other developments in the field in that it does not include material properties of the sheet metal but relies exclusively on the geometric properties of the surface. However, the method is limited by an artificial boundary condition requirement, an approximate area conservation method, and it is ineffective for surfaces with vertical flanges, i.e., areas of the surface that lie in planes perpendicular to the blank plane.

Blank development of non-developable surfaces has largely remained a finite element analysis problem. Shimada and Tada (1989, 1991) have formulated methods for such transformations of surfaces using both the finite element method and dynamic programming. In the dynamic programming approach Shimada and Tada (1991) use a two step algorithm to start a multi-stage decision process using a good initial guess, and then refine the solution to get the final two-dimensional shape. The method requires computation of strain energies and solution of stiffness matrix equations. Both the finite element technique and dynamic programming approach are computationally intensive.

2.2 Modeling of Sheet Metal Deformation Processes

Sachs (1935) is credited with creating an understanding of the mechanics of the sheet forming beginning with the successful modeling of deep drawing of a cylindrical cup. The research by Swift (1935) on deep drawing and Hill (1950) in the theory of plasticity and stability has enhanced the understanding of sheet metal deformation processes. Numerical modeling of sheet metal processes got a major thrust with the development of the Circular Grid Analysis technique and the experimental Formability Limit Diagram (FLD) developed by Keeler (1965) for strain analysis. This technique of strain analysis, uses a grid of

small circles etched on the surface of the blank sheet, which, upon deformation, causes the circles to become elliptical. The major and minor principal strains in the plane of the sheet are determined by measuring the principal axes of the ellipses. With further development of the experimental techniques by Goodwin (1968), an FLD for mild steel was obtained which, today, serves as a benchmark for the strain analysis of most automotive and appliance stampings. Theoretical formability studies and sheet forming process models have been developed to study the influence of material properties, particularly for improvements in lubrication, advanced high strength materials, and selection of process variables such as temperature and forming rate. The thrust areas of research have been the development of material constitutive relations (Wagoner, 1985; Lee and Tehrani, 1986), forming limits (Bharata et al., 1985), flow behavior (Zienkiewicz, 1984), plastic instability criteria, fracture and bending (Wang and Tang, 1986).

Modeling of the deformation processes for sheet metal focused initially on axissymmetric punch stretching. Wang (1970) developed a model based upon the incremental theory of plasticity. Lee and Kobayashi (1973) developed a Finite Element model in terms of a rigid-plastic approach called the matrix method which neglected elastic strain and proved computationally efficient in calculating plastic strains. An early attempt for an interactive design approach for process modeling was done for axissymmetric punch stretching (Nagpal et al., 1979).

With the advent of the finite element analysis technique (FEM) and the introduction of computers with high computational capability, analysis of deformation processes advanced considerably. Because sheet metal deformation is characterized by large displacements and relatively small strain rate, FEM analysis for the process is computationally inefficient. However, several aspects of sheet metal deformation behavior have been studied using the method (Tang et al., 1982; Makiinouchi, 1986; Wagoner et al., 1988; Vegter, 1988; Onate and Saracibar, 1988). Research has focused on analysis rather than design or modeling of the process, due primarily to a lack of sufficiently

robust models to predict material behavior. In the realm of computer-aided design of sheet metal surfaces, forming processes and process components, several automotive companies have developed technologies to aid in tooling and manufacturing process design (Kokkonen, 1985; Takahashi, 1985; Higashi, 1985). These tools combine the latest techniques of computer graphics for surface design, manufacturing experience and in-house technology development.

CHAPTER 3
A NEW MAPPING STRATEGY

In this chapter, an efficient algorithm is presented to determine the blank shape necessary to manufacture a surface by press forming. The technique is independent of material properties and instead uses surface geometry and an area conservation constraint to generate a geometrically feasible blank shape. The algorithm is formulated as an approximate geometric interpretation of the reversal of the press forming process. The primary applications for this technique are in preliminary surface design, assessment of manufacturability, and location of binder wrap in die-design. Since the algorithm exhibits linear time complexity, it is amenable to implementation as an interactive design aide. In Chapter 4 the algorithm is applied to two example surfaces and the results are discussed.

3.1 Material Property Independent Approach

Any tangible product is ultimately defined by its geometry; i.e., the geometric aspects of an object qualify it for any purpose. For example, an object's viability with respect to a specific functional configuration is possible only if it is geometrically feasible. It follows that, if an object can be substantiated geometrically, then it may be manufacturable. Therefore, prior to detailed design validation incorporating material properties, the geometric feasibility of a product should first be established. Unlike most physical transformations, geometric transformations are reversible. Thus if the geometry of a part and its fabrication

process are properly modeled, then the fabrication feasibility of the part can be assessed via reverse geometric transformation before detailed process analyses are attempted.

The research presented in this thesis, is motivated by this philosophy. For blank development, a qualitative indicator is sufficient at the preliminary design stage, and can be arrived at using material independent transformations. In particular, an approach similar to Chu (1983) is adopted. The methodology is independent of material properties and relies on basic geometric manipulations to derive the blank shape and other manufacturability properties.

3.2 Constant Area Transformation Algorithm

In sheet metal press forming tool design, the designer strives to achieve a uniform deformation of the blank to the final surface. The ideal deformation process is one in which the surface undergoes this transformation with no change in thickness. To determine such an ideal transformation, the tool designer would need to know where, exactly, each point on the blank lies after deformation. This requirement combined with the concept of geometric reversibility suggests the interpretation of blank development a geometric transformation problem of mapping the formed surface to a plane such that the area remains constant.

Two fundamental characteristics motivate the algorithm: 1) a procedural reversal of the forming process from the formed to the unformed state and 2) a geometric conservation of area between the two states. In effect, the algorithm transforms or "unstamps" the formed three-dimensional geometry into an initial planar shape. Variational geometry principles (Light and Gossard, 1981; Lin et al., 1981) are employed to derive a robust and efficient algorithm for the mapping. Area constraint equations limiting the degrees of freedom of points on the deformed surface are used to map each point to a feasible location in the plane containing the blank. Since the formulation is linear, the results

are accurate and computational effort increases in linear proportion to the number of surface elements that are being transformed. A geometrically feasible solution is obtained to give the designer an assessment of initial blank shape, trim line and material flow during the forming process.

The basic assumptions underlying the constant area transformation algorithm are summarized as follows:

- *The surface is represented by a grid of points (vertices) which are considered as the elemental surface entities.*
- *Elemental area entities (triangles) are formed from any three mutually adjacent non-collinear vertices.*
- *The surface is subjected to a state of plane stress only.*
- *The surface is continuous, homogenous and isotropic.*
- *Only plastic deformation is considered.*
- *The surface has no thickness.*

3.2.1 Surface geometry representation

Contemporary computer-based design tools provide several methods for generating parametric sculptured surface models. The most common representation scheme is the non-uniform rational B-splines (NURBS) surface. A B-spline surface can be represented as

$$\mathbf{S}(u, v) = \sum_{i=0}^{m} \sum_{j=0}^{n} \mathbf{P}_{ij} N_{i,k}(u) N_{j,l}(v) \qquad u, v \in [0, 1]$$

where $\mathbf{S}(u,v)$ is a three dimensional vector function of control points $\mathbf{P}_{ij}$ arranged in an $(m+1) \times (n+1)$ topologically rectangular grid and $N_{i,k}(u)$ and $N_{j,l}(v)$ are the degree k and l B-spline basis functions, respectively (Piegl and Tiller, 1987).

A well-defined homogenous triangulation of the surface is required as the input to the algorithm. Any surface $\mathbf{S}(u,v)$ can be approximated by a faceted polyhedron, defined by a (M x N) set of three dimensional vertices $V_{ij}=S(u_i,v_j)$, $i=1,\ldots, M$ and $j=1,\ldots, N$. The result-

ing polyhedron approximates the actual surface. Such a polyhedral approximation can be constructed with a specific topological structure to algorithmically exploit vertex adjacency relationships. The nature of this topology depends on the method of surface discretization. In this application, the surface is triangulated in uniform parametric intervals to form a topologically rectangular mesh, which, after mapping through $S(u, v)$ generates a uniform network. The topology of the network is constructed such that any internal vertex has exactly eight surrounding vertices as shown in Figure 1. The polyhedral surface model is stored in a data structure which distinguishes the topological and geometric information. In particular, a vertex adjacency list is created to establish connectivity between each vertex and its neighbors.

One useful characteristic of an underlying parametric surface representation is the inherent separation of the topological and geometric information. Any surface vertex in Euclidean space has a dual in the parametric domain (i.e., the *uv*-space). Since the topology of the discrete surface approximation is defined in the parameter space, operations which require adjacency information are simplified. The corresponding geometric information is thus referenced primarily for the area calculations. This separation leads to simple and efficient algorithmic implementation

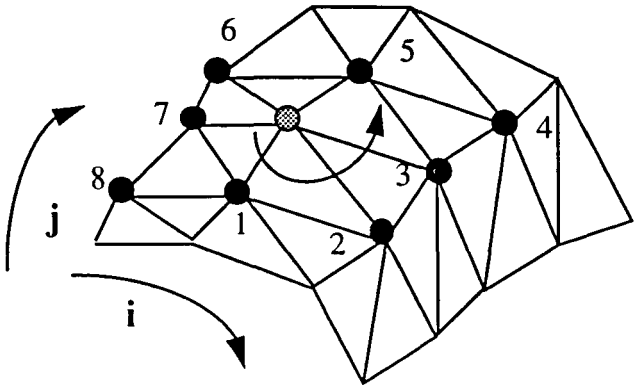


Figure 1 Discretized surface model with vertex adjacency relationships and numbering system.

3.2.2 Data structure

The information content for the algorithm is reduced to a vertex basis. The topological information for each vertex is stored in the form of a linked list containing pointers to the addresses of its neighbors. The data structure storage requirements are as follows:

Topological information	Vertex to Vertex adjacency list	8	records
Geometric information	Three-dimensional coords of each vertex	3	records
	Two-dimensional coords (to be generated)	2	records
Visit Flag	To indicate whether a vertex has been transformed	1	record
Total (*for each vertex*) =		14	records

3.3 Algorithm Basis

Any triangle in three-dimensions encloses an area on a plane. The constant area transformation is formulated such that both the topology and the area of a triangle are conserved when it is mapped from a three dimensional Euclidean space (E^3) to a two dimensional planar space (E^2). In particular, let

$$V^3 = \{V_1, V_2, V_3 \mid V_i \in \mathrm{E}^3, V_1 \neq V_2 \neq V_3\}$$

be the set of all 3-tuples which define unique triangles in E^3, and

$$P^3 = \{V_1', V_2', V_3' \mid V_i' \in \mathrm{E}^2, V_1' \neq V_2' \neq V_3'\}$$

be a set of 2-tuples which define triangles on a plane in $E^2 \subset E^3$. The constant area transformation is defined as a mapping: $\mathbf{CAT} = V^3 \rightarrow P^3$ such that for any $\alpha \in V^3$,

$$\mathbf{CAT}(\alpha) = \{\beta \in P^3 \mid Area(\beta) = Area(\alpha), Top(\beta) \approx Top(\alpha)\}$$

where *Area*() and *Top*() represent the area and topological state, respectively, of any triangle. A geometric interpretation of the constant area transformation can be summarized by the following principles:

- Given the location of two vertices V_i and V_j, a family of triangles $T(V_i, V_j, V_k)$ of area A is defined by the area locus l of the point V_k which is a line parallel to V_iV_j at a distance $h = 2A/|V_iV_j|$ from V_iV_j. (See Figure 2.)

- Given two adjacent triangles $T_a(V_i, V_j, V_l)$ and $T_b(V_k, V_i, V_l) \in E^3$ and corresponding projected locations of any three of these vertices in a plane in E^2, say V_i', V_j', and V_k', as shown in Figure 3, if the unknown common vertex $V_l' \in E^2$ is located at the intersection of area loci $l_a \leftarrow T_a(V_i', V_j', V_l')$ and $l_b \leftarrow T_b(V_k', V_i', V_l')$, then

$$(Area(V_i, V_j, V_l) = Area(V_i', V_j', V_l')) \text{ and}$$
$$(Area(V_k, V_i, V_l) = Area(V_k', V_i', V_l'))$$

These principles, although similar in spirit to those developed by Chu (1983), provide for several enhancements with respect to algorithmic implementation and computational.

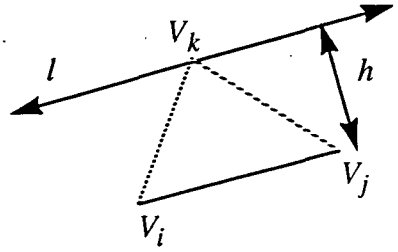


Figure 2 Locus of the third vertex of a family of constant area triangles.

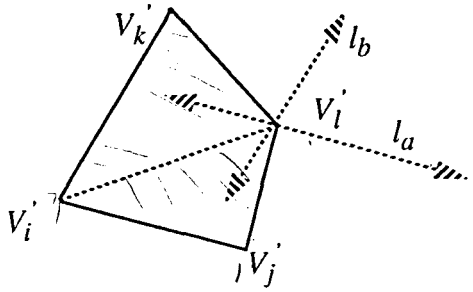


Figure 3 Area conservation principle.

accuracy. For example, as illustrated in Figure 4, Chu's constant area transformation provides an approximate solution for the location of the fourth point of a mapped quadrilateral. Given the location of three vertices V_i, V_j and V_k on adjacent triangles and their images $V_i^{'}$, $V_j^{'}$ and $V_k^{'}$ on a plane, Chu's method assumes that $Area\,(V_i, V_j, V_k) = Area\,(V_i^{'}, V_j^{'}, V_k^{'})$. Thus the area change due to the mapping of the quadrilateral (V_i, V_j, V_l, V_k) is assumed to be accounted for completely in the image of the triangle (V_k, V_j, V_l). The effect is approximated by constructing the three area loci as shown in Figure 4b. Vertex $V_l^{'}$ is taken as the centroid of the area enclosed by l_1, l_2 and l_3 as shown in Figure 4c, where

$l_1 \leftarrow T_1\,(V_i^{'}, V_j^{'}, V_l^{'})$ assuming, $Area\,(V_i^{'}, V_j^{'}, V_l^{'}) = Area\,(V_k, V_j, V_l)$

$l_2 \leftarrow T_2\,(V_k^{'}, V_i^{'}, V_l^{'})$ assuming, $Area\,(V_k^{'}, V_i^{'}, V_l^{'}) = Area\,(V_k, V_j, V_j)$

"" $l_3 \leftarrow T_3\,(V_k^{'}, V_i^{'}, V_l^{'})$

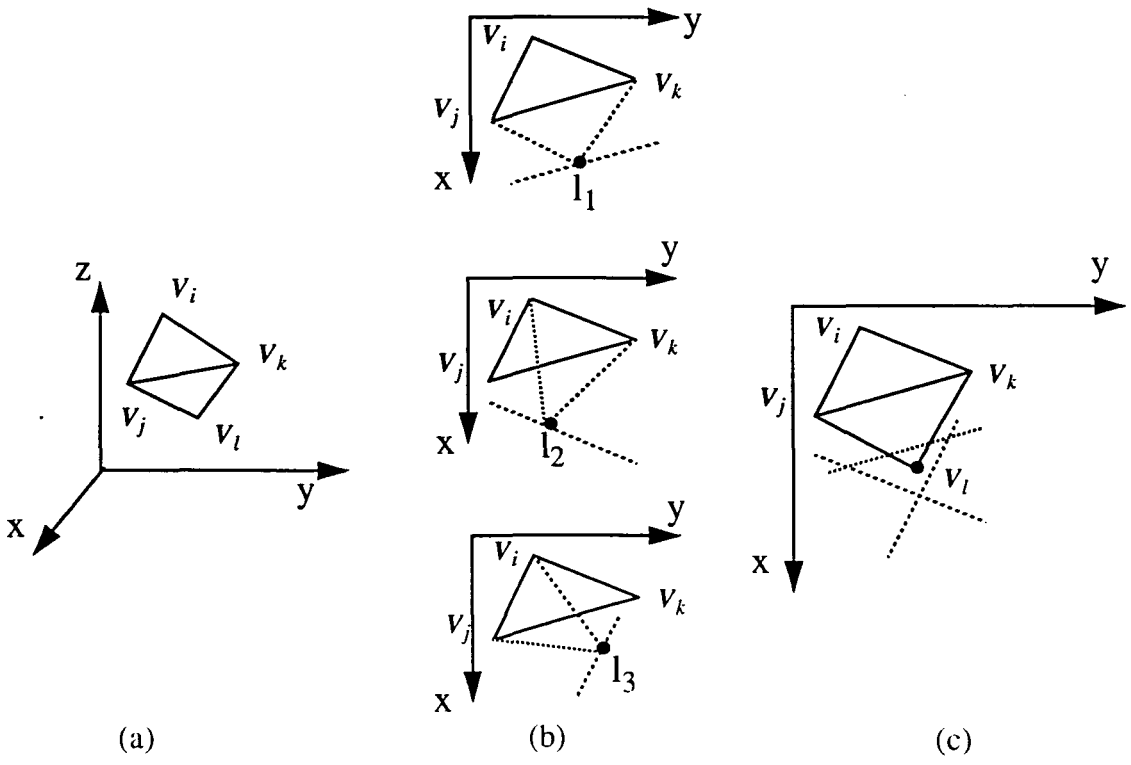


Figure 4 Methodology for mapping the fourth vertex of a quadrilateral (Chu et al., 1985).

This solution is obviously approximate due to the assumptions used to construct l_1 and l_2, and the fact that l_3 completely neglects any actual area change in the triangle (V_i, V_j, V_k). Since this formulation is this basis of the algorithm to map an entire surface, the error induced by this approximate solution is compounded because, in general, V_i, V_j and V_k are themselves calculated via the same procedure.

Another limiting aspect of Chu's formulation is the requirement of imposed boundary conditions necessary to initialize the algorithm. An orthogonal frame of reference along approximate planes of part symmetry must be established on the surface prior to the mapping. This frame is chosen such that the image of the vertices which lie on them can move only along axes formed from the intersection of the blank plane and the symmetry planes, as shown in Figure 5. Boundary conditions are user-defined, such that the boundary vertices of any two sides of the surface are fixed and identified along the reference frame. Both the symmetry reference frame and the boundary conditions impose restrictions on the material flow which, in general, do not reflect an accurate model of the forming process. The present work, however, derives its methodology from the forming process directly. The surface is assumed to be constrained equally along all edges to provide a restricted inflow of material. In the geometric context, this equates

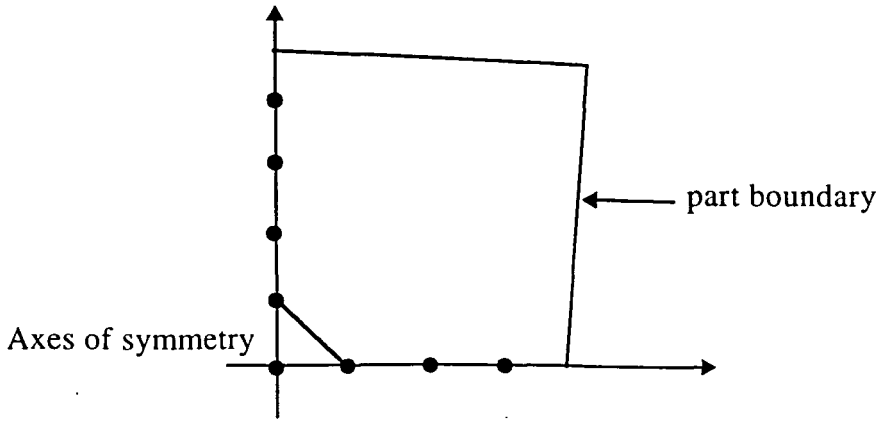


Figure 5 Boundary conditions for mapping (Chu et al., 1985).

to a "boundary-less" or free-form deformation, since forces (equal along all edges) are not interpreted geometrically. This characteristic of the algorithm is described in detail on the following two sections.

The following terminology is used in the remainder of the chapter to facilitate the description of the constant area transformation algorithm:

N	Total number of vertices on the surface
V_{ij}	Address of a specific vertex, i and j correspond to surface parametric direction u and v.
$Adj(V_{ij})$	Vertex adjacency list for each V_{ij}
$Vert3D(V_{ij})$	Three-dimensional coordinates of vertex V_{ij}
$Flag(V_{ij})$	1 - if vertex V_{ij} is transformed, 0 - otherwise
$Vert2D(V_{ij})$	Two dimensional coordinates of the vertex V_{ij}
Primary Neighbors	Vertices in $Adj(V_{ij})$ which are topologically adjacent to V_{ij} in parameter space, i.e., $V_{i-1,j}$, $V_{i+1,j}$, $V_{i,j-1}$, $V_{i,j+1}$
Secondary Neighbors	Other vertices in the $Adj(V_{ij})$ list
Visit_List	List of vertices to visit

3.4 Transformation Initialization

Without lack of generality, the algorithm assumes that the blank plane is the global XY plane and the punch travels parallel to the Z axis. Since thinning is assumed negligible in an ideal forming process, the point of initial punch contact on the blank will most likely lie in the vicinity of the formed surface point which is furthest from the blank plane. Thus, the vertex with the largest Z-value, V_{max}, on the interior of the surface is taken as the reasonable starting point for the mapping. To reverse the forming process, "un-forming" of the surface from this initial point creates the effect of reversing the flow of material from the final state to the initial state. (In actual formed products distinct point peaks on the formed surface may not exist, instead a plateau of surface points at the maximum height can be

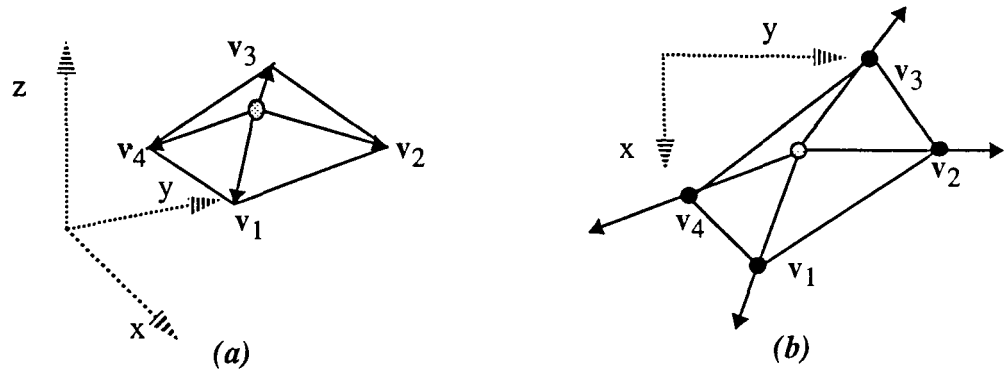


Figure 6 Transformation of V_{max} and the surrounding primary neighbors by preserving the length of line of vectors v_1, v_2, v_3 and v_4.

found.) To mimic the uniform inflow of material to form a final surface, a uniform reverse outflow of area is formulated. This flow is achieved by allowing the triangular elements to transform in a concentric manner starting with the vertices immediately surrounding V_{max} as shown in Figure 6. The transformation is initiated by projecting V_{max} parallel to the Z-axis, onto the blank plane. To seed the algorithm, the location of at least two additional vertices (adjacent to V_{max}) on the blank plane are required. However, the boundary-less deformation assumption implies a uniform flow of area toward all surface edges. Thus, the locations of the four primary neighbors of V_{max} on the blank plane are required. To map the primary neighbors, area preservation techniques cannot be applied, since V_{max} is the only vertex identified on the plane. Therefore, a length of line preservation procedure is adopted as shown in Figure 6. Unit vectors, formed between V_{max} and each of its primary neighbors are projected on the blank plane parallel to the Z-axis. The locations for the primary neighbors on the blank plane are approximated by scaling the two-dimensional vectors to their original three-dimensional length.

Algorithm: Transform initial primary neighbor vertices (constant time)

Input: V_{max}

Output: primary neighbor Vert2D images of V_{ij}

{

calculate the primary neighbor vectors
calculate the lengths of the vectors
normalize the vectors and project on the blank plane
scale 2D vectors to 3D lengths
calculate primary neighbor vert2D point images
return point images
}

The remaining vertices of the surface are then transformed by visiting each vertex and inspecting its neighbors. Each vertex is initially checked for mapping status. If the vertex has not been mapped, the algorithm searches the vertex adjacency list to determine whether at least three adjacent vertices have already been mapped. If the criterion is met, the routine returns that the vertex *CAN BE MAPPED*. Otherwise the function returns that the vertex *CANNOT BE MAPPED*.

Algorithm: Vertex map feasibility (constant time)

Input: V_{ij}
Output: vertex map feasibility
***MAPPED_NEIGHBORS*=0**
{
for all $V_k \in Adj\,(V_{ij})$
{
if (Flag(V_k*) = VISITED)*
{
add V_k *to Mapped_list(*V_{ij}*)*
increment ***MAPPED_NEIGHBORS***
}
}
if (*MAPPED_NEIGHBORS* ≥ 3)
{
if at least three consecutive $V_k \in$ *Mapped_list* (V_{ij}) *are VISITED*
return vertex CAN BE MAPPED
else return vertex CANNOT BE MAPPED
}
else return vertex CANNOT BE MAPPED
}

A vertex which meets the criterion for mapping is mapped according the area conserving hypothesis. Depending upon the number of starting peaks, there may be more than three mapped neighbors surrounding the vertex which is queried. The solution methodology for the various cases are discussed in Chapter 6.

3.5 Vertex Mapping

This is the kernel of the algorithm. Prior to this step, all the vertex manipulations are done in the topological space. In this segment of the algorithm, the geometric data is accessed and the area calculations are performed as described in section 3.2.

Algorithm: Map vertex (time complexity = O(n))

Input: V_{ij} and three neighboring mapped vertices.
Output: Vert2D image of V_{ij}
{
calculate the area enclosed by the vertices in three-dimensional space
calculate 2D triangle base lengths
calculate area locus of V_{ij} from each of the two adjacent triangles
calculate Vert2D(V_{ij}) = intersection of the two loci
return Vert2D(V_{ij}) coordinates
}

3.6 Mapping Remaining Vertices

After the first vertex and its primary neighbors are mapped, the remaining vertices are scanned by generating a visit list which is initialized with the addresses of the primary neighbors. Primary neighbors of each element of the visit list are queried for mapping via the *Vertex map feasibility* algorithm. Those which can be mapped, are mapped and appended to the visit list. The algorithm proceeds in this manner until the visit list is exhausted. The algorithm structure follows.

Algorithm: Map remaining vertices (time complexity = O(n))

Input: Initialized visit_list with primary neighbors of (V_{max})
Output: Vert2D coordinates of all vertices of the surface
for all $V_k \in$ visit_list
{
for all primary neighbors of V_k
{
query = Vertex Map Feasibility(Adj(V_k))
if query = CAN_BE_MAPPED
{
Map_the_Vertex(Adj(V_k))
add Adj(V_k) to the visit_list
}
}
}

CHAPTER 4
SINGLE PEAK MAPPING

Two applications of the algorithm to surfaces with single peak points are analyzed in this chapter.

4.1 Mapping Examples

Two example applications are presented which demonstrate constant area transformation of surfaces with single peak vertices. Computation times reported reflect implementation on a Silicon Graphics Indigo workstation with 48MB of RAM.

4.1.1 Example 1: Bezier surface

The fan shaped bicubic Bezier surface shown in Figure 7 was represented by a 40 by 40 parametric subdivision to yield 1600 vertices on the tessellated surface. The mapping was performed and the result is shown in Figure 8. The mapping of the surface was well defined and showed fairly uniform material flow over the entire surface. An increase of the parametric sampling produced the same result in slightly greater detail. The computation times for two surface discretization densities are shown in Table 1. As expected, the computation time grew in linear proportion to the number of vertices used to represent the surface.

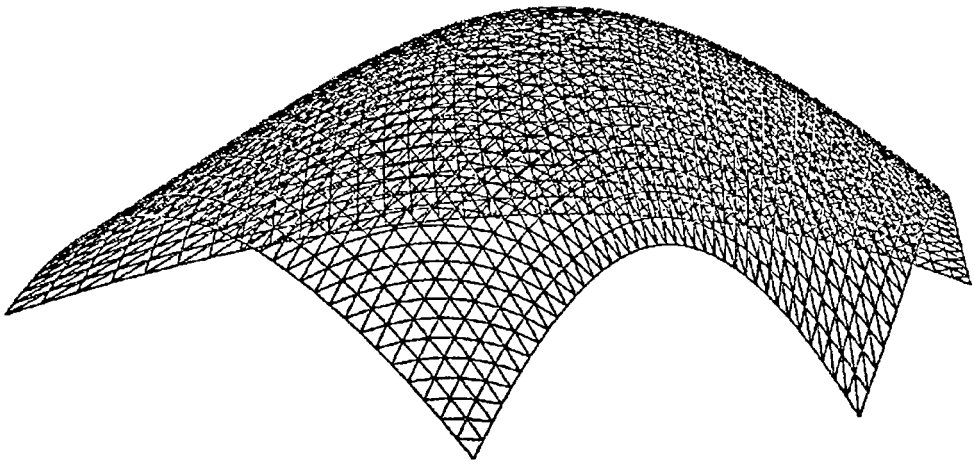


Figure 7 Example 1: Bezier Surface.

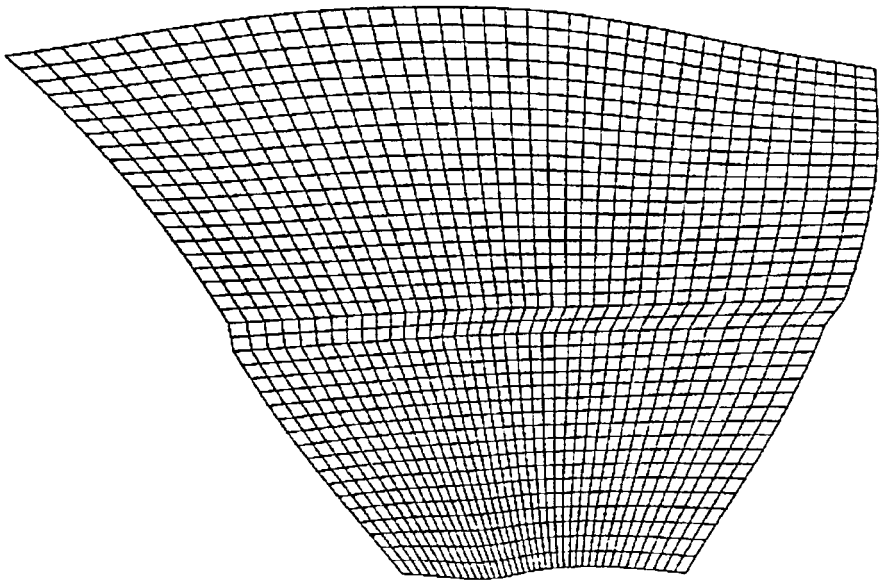


Figure 8 Example 1: Mapped surface, 40 by 40 parametric subdivision.

4.1.2 Example 2: B-spline surface

This example is a more complicated B-spline surface of a toy model sports car body. This surface is characterized by a single peak, flanged edges and multiple peaks at local maxima as shown in Figure 9. This surface was chosen to study the behavior of the algorithm in the regions of flanges which the previous geometric method developed by Chu (1983), could not handle. A 40 by 40 grid of surface points was generated and the algorithm was applied to this surface definition. The result is shown in Figure 10. The algorithm produced very interesting results for the surface. On visual inspection, the material outflow correlated with the probable material inflow during actual forming operation. The mapping showed a slight bunching of the grid elements in the lower most edge of the surface near the front of the car body. This corresponds to a region of complex curvature on the original surface, thereby signifying that the edge will have severe compressive forces acting on it, resulting in possible wrinkles. The remaining portion of the surface showed no severe area distortion. The computational results for the surface are summarized in Table 1.

Table 1 Surface Mapping Results

		Example 1: Bezier Surface		Example 2: B-spline surface	
	Discretization	40 x 40	60 x 60	40 x 40	60 x 60
Area Calculation	Computation Time (sec)	1.22	1.90	1.35	1.60
	3D Area	8.52	8.52	17.66	17.70
	2D Area	8.52	8.52	17.66	17.70
	Difference (%)	0.0012	0.0005	0.0140	0.0072

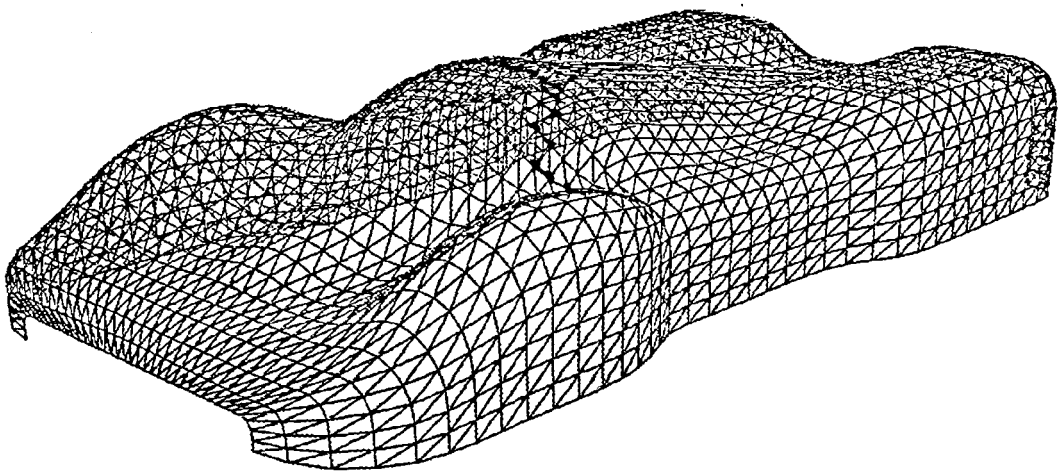


Figure 9 Example 2: B-Spline surface.

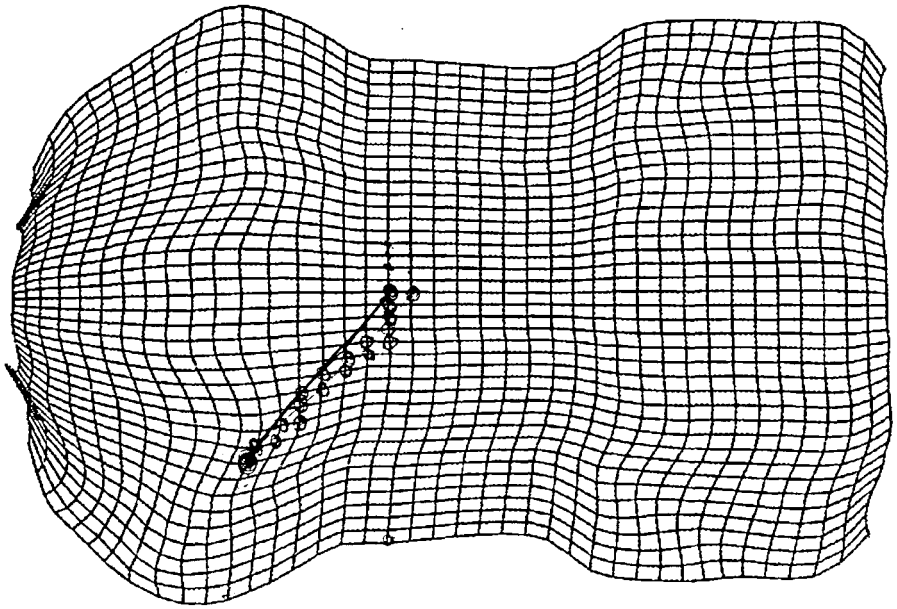


Figure 10 Example 2: Mapped surface, 40 by 40 parametric subdivision.

4.2 Robustness Issues

Numerical degeneracy arises during computation of an intersection point when the generating area loci are nearly parallel to each other. Since the basic methodology of the constant area transformation algorithm is propagation of areas from the center, any numerical error propagates, resulting in the failure of the algorithm. Although, this is not a deficiency of the algorithm, the resulting blank can be influenced considerably by the ability to detect and accommodate this computational error.

The problem is rooted in the interaction of approximate numerical and exact symbolic data (Hoffman, 1989). According to Hoffman (1989), geometric objects belong conceptually to a continuous domain, yet they are almost always analyzed by algorithms which employ discrete computation. The present algorithm treats a very large discrete domain as though it were a continuous domain and results in the occasional failure of the algorithm.

To enhance robustness, the problem of nearly parallel line intersection is solved by making interdependent logical decisions, giving priority to the numerical data, and slightly

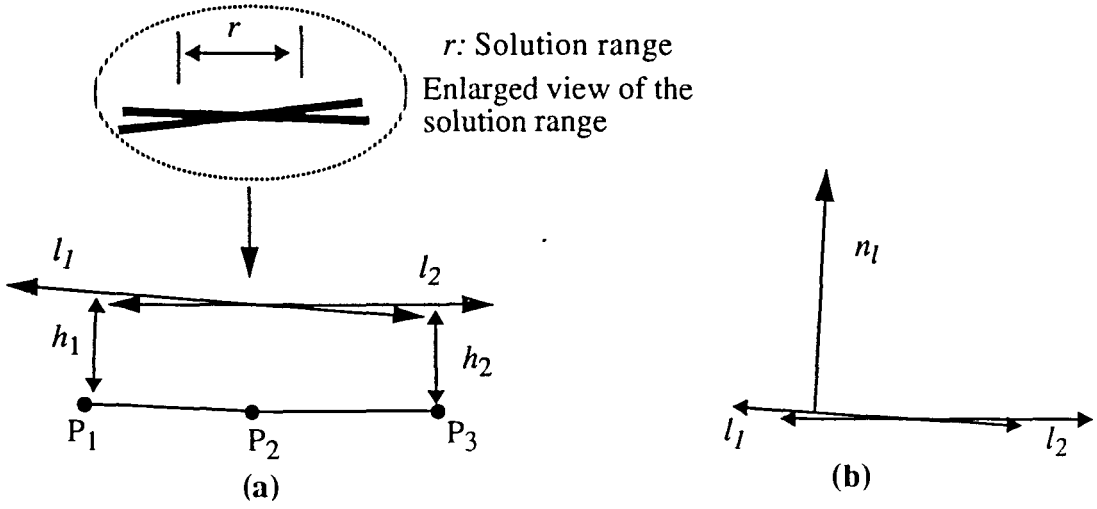


Figure 11 (a) Condition for numerical degeneracy during loci intersection calculation. (b) Dot product of the normal to l_1 and l_2.

altering the meaning of the symbolically represented topological problem data. The intersection problem is illustrated in Figure 11.

4.2.1 Common normal method

Let n_l be the normal to locus l_1 as shown in Figure 11b. Then the scalar product $n_l \cdot l_2$ (denoted by the condition number ε) is an indicator of the degree of parallelism between loci l_1 and l_2, i.e., if $n_l \cdot l_2 \approx 0$, the two loci l_1 and l_2 are nearly parallel to each other. Since the computation involves floating point calculation, the intersection solution will lie in the range r as shown in Figure 11a. This problem is avoided as shown in Figure 12. For a condition number ε below which the calculation was ill-conditioned or unstable, the intersection point is assumed to lie a common normal at the vertex P_2 common to the two bases at a distance h calculated using the area loci method as explained in CHAPTER 3. Here, the local topology of the two lines is altered to form a single line such that the length is the sum of the two individual base segments and the common slope is the average of the two slopes. The common normal is defined perpendicular to this slope.

Tests with various values of ε show that on increasing the discretization of the design surface, the ε can be reduced to a small value, thus reducing the influence of the slight al-

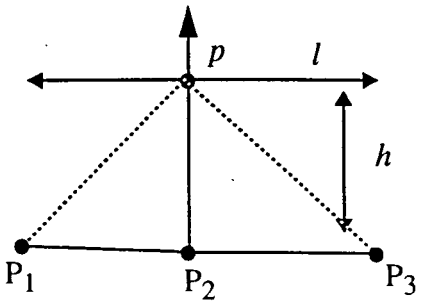


Figure 12 Common normal method for avoiding numerical degeneracy.

teration of the problem format. Since, the algorithm is linear time, the common normal technique is computationally inexpensive. This is a very attractive feature for the algorithm's implementation as a general design aid.

CHAPTER 5
MULTI-PEAK MAPPING

For surfaces characterized by more than one peak of nearly the same height, the mapping strategy is substantially more complex. Some of the important concepts in multi-peak mapping are discussed in this chapter.

5.1 Transformation Initialization

The transformation initialization strategy must be generalized to deal with multi-peak mapping. The key issue here is the starting vertex determination methodology.

5.1.1 Starting points

For more than one peak, the mapping start point is not readily intuitive as illustrated in Figure 13. A cross section of a surface with two peaks P_1, P_2 is shown in Figure 13a. In actual forming, depending on the direction of punch travel, the three points P_1, P_2 and G will make contact with various parts of the punch and die at approximately the same time. Furthermore, if the surface is viewed from the opposite direction to the original line of sight, the surface reduces to a single peak surface with the single peak G as shown in Figure 13b. This implies that a change in the viewing direction could result in different set of starting points for the transformation. For the case shown in Figure 13, the mapping start points would include all the points P_1, P_2 and G.

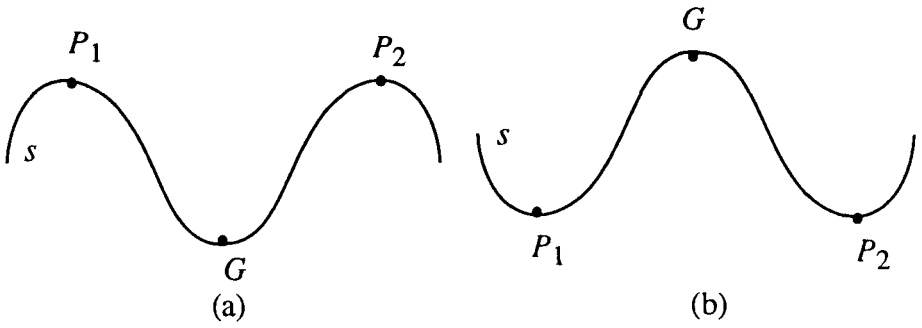


Figure 13 (a) Surface with two peaks P_1 and P_2 (b) Same surface inverted

5.1.2 Convex hull technique

The initial data set is not restricted to a collection of single vertices of the surface approximation grid. For cases in which, there are plateaus of vertices at approximately equal Z-value, the start points are chosen such that they are the centers of approximate convex hulls, in topological space, of the plateau vertices. Since the vertices inside the convex hull are assumed to have the same constant Z values, they are mapped to the plane with no change in the area contained between them.

5.2 Moving Front Area Calculation Techniques

In the general multi-peak implementation, cases will arise in which more than the minimum of three mapped neighboring vertices exist for mapping. In other words, the vertex to be mapped is part of one or more moving fronts and its position is affected by its mapped neighbors due to the area constraint. Extra vertices are eliminated to reduce the problem to a three-vertex case by viewing the vertex and its neighborhood in topological space. There are five specific cases depending on how many mapped neighbors

surround the vertex. In all the cases, exact solutions for the mapped two-dimensional coordinates are determined. The solution procedure for the specific cases are discussed here.

Four adjacent neighbors: Consider the case as shown in the Figure 14. For vertex *P*, neighbors *7, 0, 1* and *2* are already mapped. Overhanging neighbor Vertex 7 is eliminated from the area loci calculations since it is not part of any area enclosing triangle in the topological neighborhood of *P*. The elimination of the overhanging neighbor prevents this imbalance. An alternative approach is to consider the area contribution of triangle T(7, 2, 0) in addition to T(0, 2, 1). However, the experimentation with such a formulation revealed that it can induce an undesirable imbalance in the overall map due to the non-uniform attractive and repulsive area components. The problem is thus reduced to the three-vertex case which is solved using the principles explained in the preceding sections.

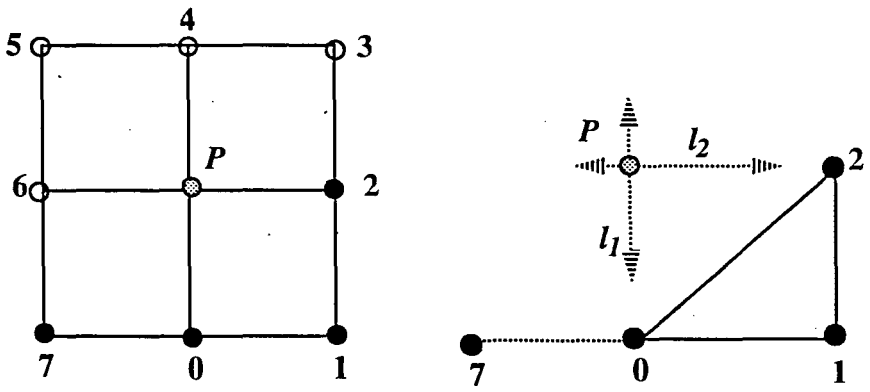


Figure 14 Four neighbor case. Vertices 7, 0, 1 and 2 are mapped.

Five adjacent neighbors: Two types of neighborhood arrangements are possible as shown in the Figure 15 and Figure 16. In both cases, all the surrounding mapped neighbors are involved in the area calculation since they constitute area enclosing triangles in the topological neighborhood of *P*. As shown in Figure 15, the five-vertex case reduces to a three-vertex case with adjacent triangles $T_1(6, 0, P)$ and $T_2(0, 2, P)$.The nominal area loci

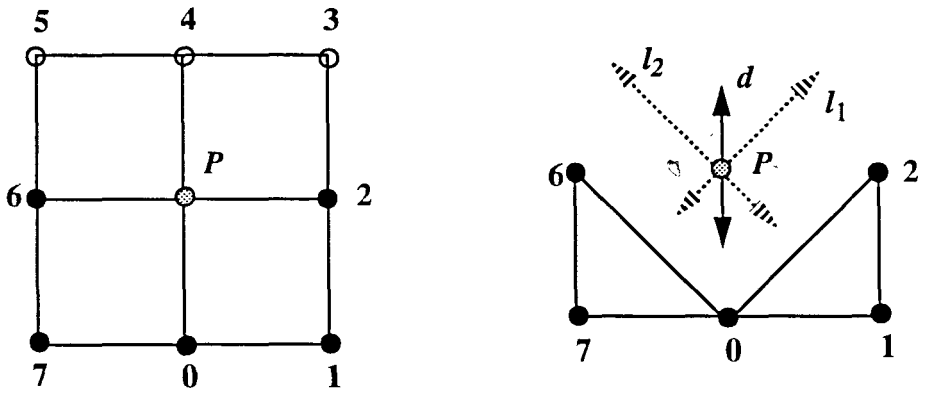


Figure 15 Five-neighbor case. Type A, reduces to a three-vertex case by considering vertices 6, 0 and 2.

formulation is modified to account for the change in total area due to the projection of triangles $T(6,7,0)$ and $T(0,1,2)$. For example, the area used to determine area locus l_2 is taken as,

$$A_{l_2} = Area(6,0,P) + [Area(6,7,0) - Area(6',7',0')]$$

The formulation for area locus l_2 is modified in a similar manner. This modification ensures that constant area is maintained between P and its mapped neighbors.

The second type of five-neighbor case is shown in Figure 16. This case is more complicated since it involves included areas in different planes. n this case, a pseudo vertex (V_p) is calculated to reduce this case to a similar variation of the standard three-vertex problem. Vertex V_p is formed by the intersection of the lines connecting vertices 0, 3 and 7, 2 respectively as shown in Figure 16. Since the vertices are located in different planes in E^3 space, a length of line approach is adopted to map the vectors to the same plane as explained in CHAPTER 3, Section 3.4. The three vertices 7, V_p and 3 form the vertices for the three-

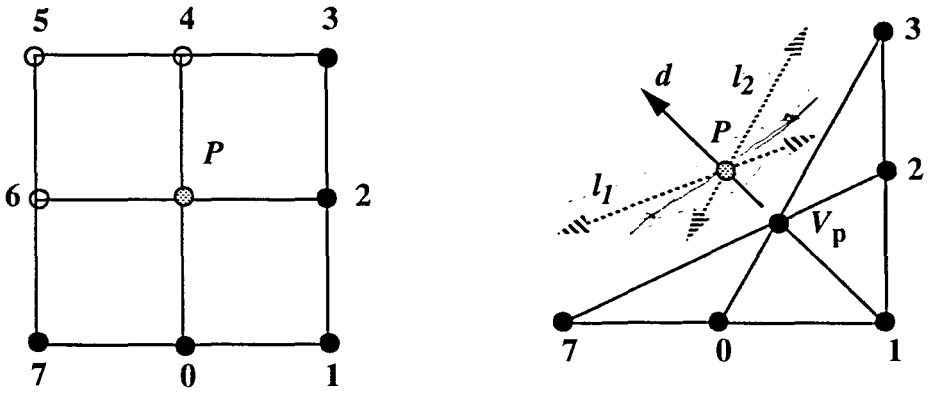


Figure 16 Five neighbor case. Type B, reduces to a three-vertex case by considering vertices 7, V_p, and 3.

vertex problem. The area loci calculation is modified to take into account, the areas enclosed by the previously mapped neighbors. For instance, area used to determine area locus l_1 is:

$$A_{l_1} = Area(7, V_p, P) + (\{Area(7, 0, V_p) + Area(0, 1, V_p)\} - \{Area(7', 0', V'_p) + Area(0', 1', V'_p)\})$$

Six adjacent neighbors: This situation is reduced to a five-vertex case by eliminating one overhanging neighbor for the same reason as the four adjacent neighbors case. Then the five-vertex technique is employed as shown in Figure 17.

Seven adjacent neighbors: This is similar to type A of the five-vertex case shown in the Figure 15. The case is easily reduced to a three vertex case by considering two adjacent triangles enclosed by vertices 5, 0, 3 and sharing P as shown in Figure 18. To preserve area, the net area components of the triangles indicated by the shaded areas are added to the area loci calculation for determining vertex *P*.

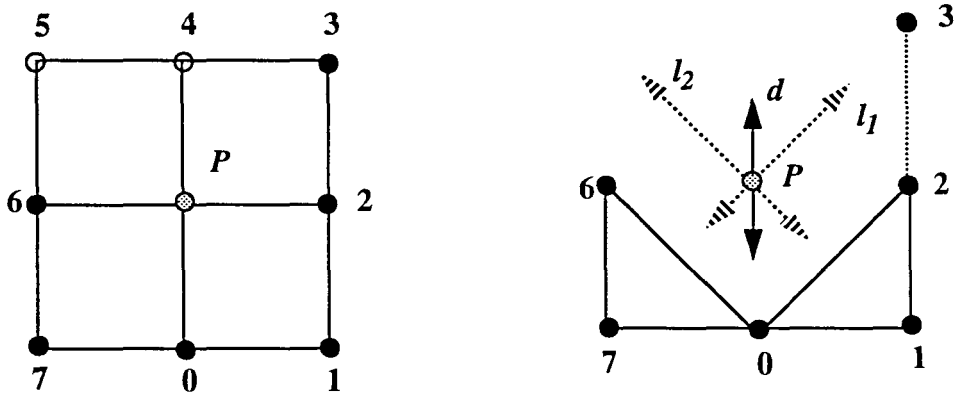


Figure 17 Six neighbor case. Reduced to a five neighbor case.

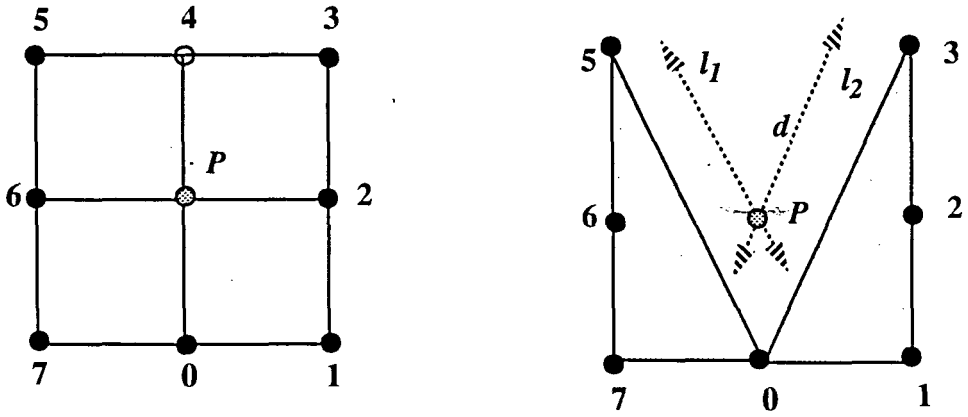


Figure 18 Seven neighbor case. The problem is reduced to a three vertex case by considering vertices 5, 0 and 3.

5.3 Multi-Peak Implementation

Multi peak surface blank development is a more complicated extension to the mapping algorithm explained in the preceding chapters. Merging fronts play a important role in the material propagation towards the edges. Another important factor is the various stages of merging. In common products like industrial panels among others, sheet metal is usually

formed to shapes containing flat surfaces at different Z values. During the punch travel in the forming cycle, in lieu of single points, small areas on the blank corresponding to these flat surfaces come in contact with the punch or die. Every step where an area of the blank is directly involved in the deformation can be assumed to be a stage. Rigorous theories need to be established to correlate the merging of the fronts and the effect of stages in the merging process. It is the view of the author that a volume conservation technique would hence be a more realistic approach to the blank development since change in thickness occurs during any press forming process owing to its less than ideal performance. In the present method, the strain energy is not taken into account for the distortion of the lengths between the vertices representing the surface. In reality all atoms tend to follow the path of least energy to change from one state to another. The application of this concept could lead to a more robust solution.

CHAPTER 6
MANUFACTURABILITY CONSTRAINTS

Two evaluation functions for determining the manufacturability of a surface by press forming are discussed. The first is a visibility condition for the mating surface of the die-set used for the forming process. The second function is an evaluation of the geometric strain on the design surface. The strain calculation makes use of the constant area transformation and is performed by assuming a pure homogenous mode of deformation (Sowerby et al., 1982; Chu, 1983).

6.1 Visibility Criterion for the Mating Surface

The mating surface of the punch and die defines the surface manufactured using the die-set. A well defined mating (parting) surface ensures that the forming operation can be performed without any interference between the punch and the die. This problem can be evaluated by employing the concept of visibility. Two points are visible to each other if they can be joined by a straight line (Shin, 1986). On looking along the line of punch travel (the direction of stamping), if all the points on the mating surface of the die (or punch depending on the direction of sight) are visible to the corresponding points on the punch (or die), then the parting surface can be considered well defined. In this research, local visibility refers to being able to completely see the object along the direction of punch travel. A schematic of two die-sets is shown in Figure 19. Figure 19a shows a case where there is

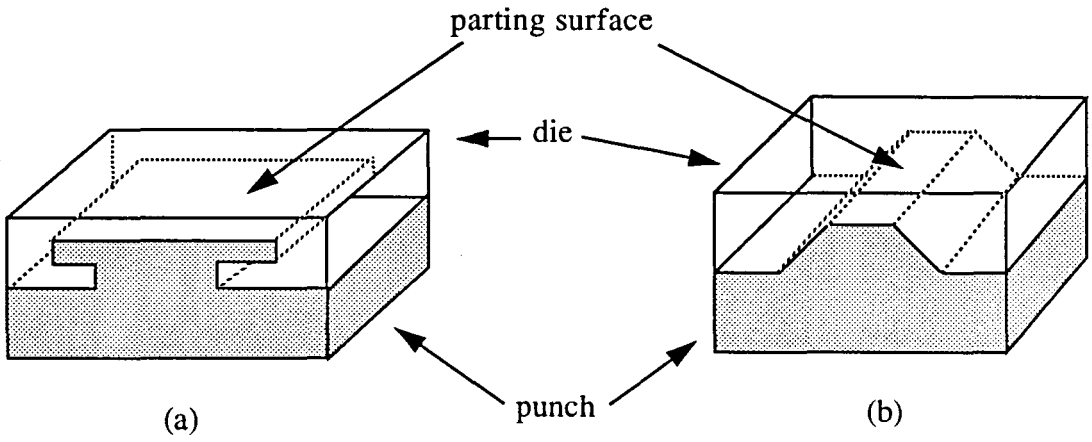


Figure 19 Die-set schematics showing the parting surface profile. (a) Interference at the parting line between die and punch. (b) parting surface well defined.

interference between the die and punch, and a well defined parting surface is seen in Figure 19b. The formability of a surface is hence directly dependent on a well defined parting surface between the die and punch.

Since the die and punch mating profiles are determined from the product surface model, it would be sufficient to study the design surface to determine the feasibility for press formability of the surface. A geometric algorithm based on a visibility criterion is formulated for determining the press formability of a surface using principles of computational and spherical geometry. This algorithm is based on the work of Chen and Woo (1992) and Chen et al. (1992).

Visibility is a useful concept which is extensively used in computational geometry. The gaussian map (GMap) is employed to determine the visibility of the surface.

6.1.1 Gaussian map

Consider a planar face f with normal n_f. The set of directions from which the surface is locally visible is v such that $\{ v | v \cdot n_f \geq 0 \}$. The gaussian map (GMap) of a surface is defined as a mapping of the given surface onto a unit sphere by translating each of the surface

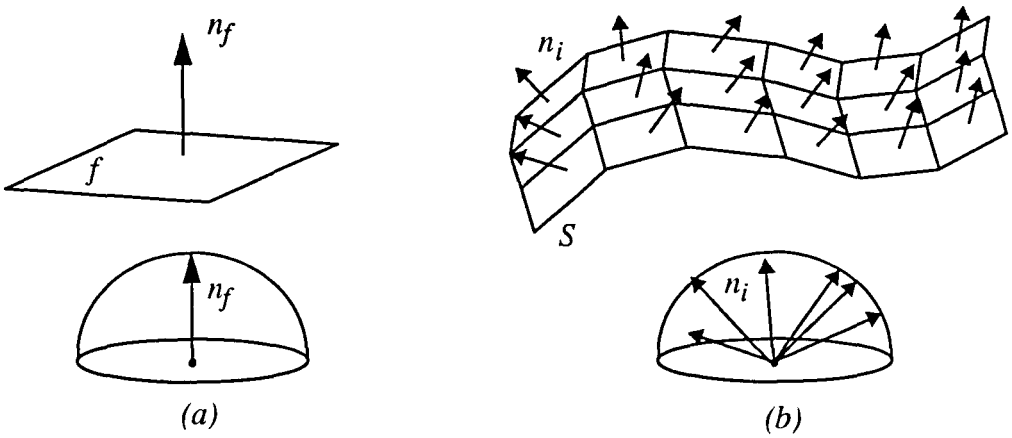


Figure 20 Surface normals and the representation on the Gaussian sphere for a planar surface and a planar approximation to a curved surface *S*

unit normals of every point on the surface to the sphere origin as shown in Figure 20a and b. A curved surface can approximated by a polyhedron, each of its faces being a planar surface. The intersection of the unit sphere with the normals results in the gaussian map of the curved surface as shown in Figure 20b. The GMap can be expressed as:

$$GMap\,(f) \;=\; \{n_f\} \tag{6.1}$$

Each polygon of the curved surface approximation corresponds to a point and the curved surface corresponds to a region on the GMap.

6.1.2 Visibility map

The visibility map (VMap) is defined as the set of unit vectors v, such that the surface is locally visible if the direction of sight is aligned in the direction v. Mathematically, the visibility map of a planar face can be expressed as the hemisphere $H = \{v \mid v \cdot n_f \geq 0\}$. In other words, VMap for a surface f is the hemisphere represented by the unit normal n_f on the unit sphere as shown in Figure 21. More generally, the visibility map of a polygonal surface is defined as the intersection of hemispheres that correspond to points in the GMaps

of the surface. Consider a polyhedral surface S, $S = \{f_1, f_2, f_3, ...f_n\}$. Let n_{f_i} be the unit surface normal of face f_i. If v_i is in VMap(S), then,

$$v_i \cdot n_{f_1} \geq 0 \Rightarrow v_i \in H_1$$
$$v_i \cdot n_{f_2} \geq 0 \Rightarrow v_i \in H_2$$
$$v_i \cdot n_{f_3} \geq 0 \Rightarrow v_i \in H_3$$
$$\cdot \qquad \cdot$$
$$v_i \cdot n_{f_n} \geq 0 \Rightarrow v_i \in H_n$$

where $H_i = \{u | u \cdot n_{f_i} \geq 0\}$. The visibility map of the surface S is defined by:

$$VMap(S) = \{v | v \cdot n_{f_i} \geq 0, i = 1,.....,n\} = H_1 \cap H_2 \cap H_3 \cap \ldots \cap H_n.$$

Recall that the Gaussian map of the surface is defined as: $GMap = \{nf_i | i = 1,.....,n\}$. Let $H = \{u | u \cdot v \geq 0, v \in VMap(S)\}$. Then $GMap(S) \subseteq H$.

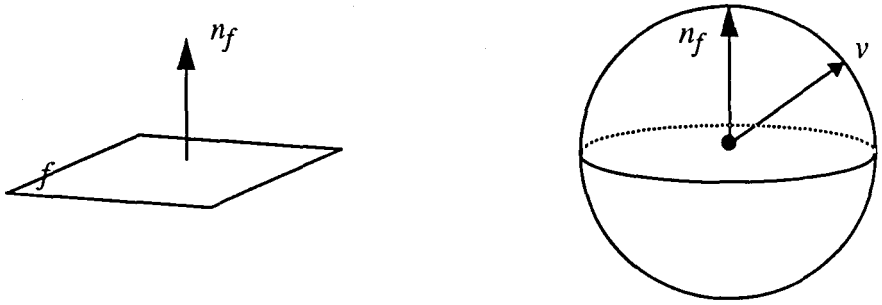


Figure 21 Plane surface f and the Visibility Map VMap(f)

6.1.3 Duality relation between VMaps and GMaps

The VMap of a surface is closely related to its GMap, which represents each normal of the surface by a point on the Gaussian sphere. If all points on the surface are to be visible, the VMap of the surface will consist of all unit vectors v that deviate from each vector in the GMap by an angle of at most 90°. The VMap of a surface is therefore the intersection of VMaps of all the points on that surface. Since the VMap of a point is a hemisphere on

the Gaussian sphere, the VMap of a surface is the intersection of hemispheres - a spherically convex polygon. Furthermore, since a vector v in the VMap deviates from each vector in the GMap by an angle of 90° at the maximum, the GMap must be a subset of the hemisphere defined by the vector v such that

$$\{u : (u \cdot v) \geq 0 \quad \text{where} \quad v \in VMap\} \tag{6.2}$$

In other words, $GMap(S) \leq H$. By decomposing a sculptured surface into a set of planar polygons $f_1, f_2, ..., f_n$, we can compute whether the surface can be manufactured by press forming by analyzing the GMaps of the polygons f_i. Let GM(f_i) and GM(f_j) denote the GMaps of polygons f_i and f_j. If GM(f_i) and GM(f_j) are subsets of a hemisphere defined by $\{u|\ (u \cdot w) \geq 0, |u| = 1\}$ then f_i and f_j are locally visible and can stamped by a punch and die aligned with the vector w. If GMaps of all the planar polygons f_i are contained in the same hemisphere, then the surface is locally visible along the orientation w and can be stamped in the single orientation. Therefore, the visibility criterion for press forming of a sculptured surface is to find a hemisphere that contains all the GMaps of its elemental areas (triangles). It should however be noted that the formulation is approximate since the planar polygons themselves constitute an approximation of the surface. Since the GMaps of these triangles are spherical points, this leads to the formulation of a geometric problem on the sphere.

6.1.4 Hemisphericity test

Given n points on the sphere, determine whether all the points are contained in a hemisphere.

Procedure:

Step 1. Consider the outward (or inward) pointing normals n_{f_i} of the polyhedron. Form the GMap for the surface using the normals. Without lack of generality, it is assumed that no points lie on the equator by performing initial rotations. For the gaussian sphere, use

central projection to obtain two sets of points, P^+ and P^- on the plane $Z = 1$ as shown in Figure 22. Define $P = \{P_1, P_2, P_3, \ldots P_n\}$ the set of points on the sphere, then

$P^+ = P \cap H^+$ where $H^+ = \{(x, y, z) / z > 0\}$ (*northern hemisphere*) and

$P^- = P \cap H^-$ where $H^- = \{(x, y, z) / z < 0\}$ (*southern hemisphere*)

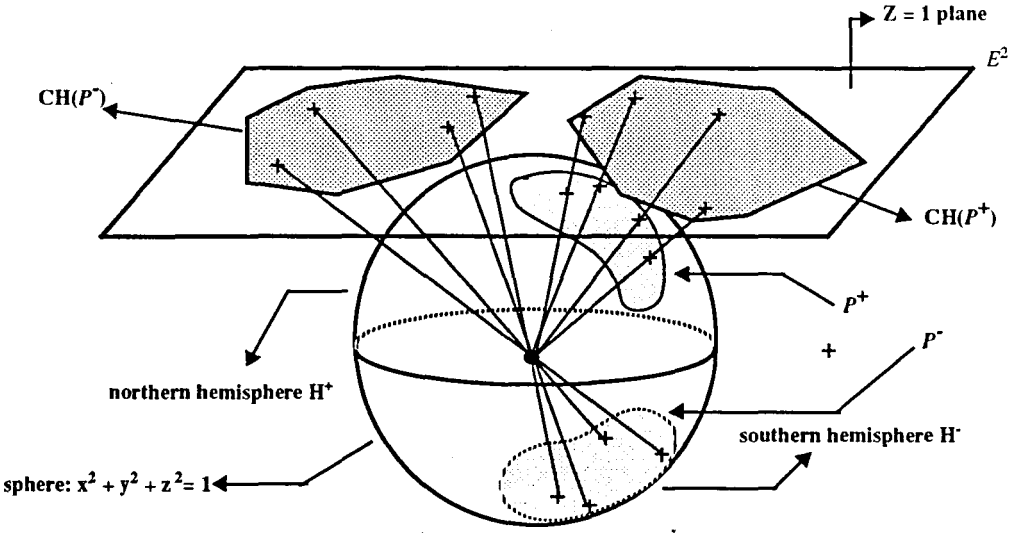


Figure 22 Use of central projection to map points on the gaussian sphere to the Z=1 plane

Central projection maps a pair of antipodal points on the unit sphere S^2 to a point in a two-dimensional projective plane P^2, which consists of E^2 augmented by a line at infinity. By emitting a ray from the origin, a point (x_1, x_2, x_3) in S^2 with $x_3 > 0$ is mapped to $(x_1/x_3, x_2/x_3, 1)$ in the plane $x_3 = 1$, corresponding to the point $(x_1/x_3, x_2/x_3)$ in P^2. The antipode of point (x_1, x_2, x_3), $(-x_1, -x_2, -x_3)$, is also mapped to $(x_1/x_3, x_2/x_3)$ in $x_3 = 1$ (and thus $(x_1/x_3, x_2/x_3)$ in P^2). A point (x_1, x_2, x_3) with $x_3 = 0$ is mapped onto the line at infinity in P^2.

Step 2. Construct two planar convex hulls $CH(P^+)$, $CH(P^-)$ for P^+ and P^-. The hulls can be found out using a convex hull algorithm, for instance, Graham's algorithm (Graham, 1972) in O(nlogn) time.

If either P^+ or P^- is ϕ, it implies that P is hemispherical (if P^-: ϕ, P^+ is the hemisphere and vice-versa). The procedure is terminated.

Step 3. Test if $CH(P^+)$ and $CH(P^-)$ intersect. This can be determined in $O(n)$ time using the algorithm proposed by O'Rourke et al. (1982).

Mathematically, P is hemispherical if

$$int\,[CH(P^+) \cap CH(P^-)] = \phi \tag{6.3}$$

otherwise P is not hemispherical. If the two convex hulls do not intersect, a separating line can be identified between $CH(P^+)$ and $CH(P^-)$. This can be done using Megiddo's algorithm (Megiddo, 1983; Shamos, 1978). A great circle results from the projection of the separating line on the sphere as shown in Figure 23. The plane containing the great circle determines the parting line for the hemisphere that contains all the Gaussian points. Since all the surface normals are contained in the hemisphere, the normal dual of the hemisphere determines the local visibility direction for all the surface points. In other words, for press

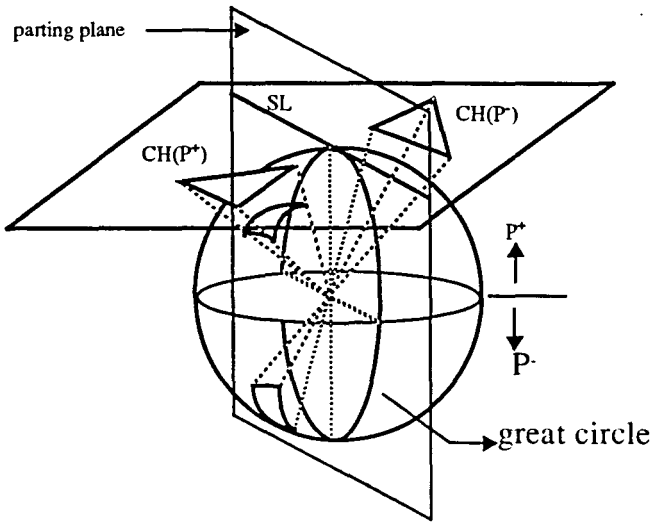


Figure 23 Hemisphericity test: Finding the parting line and the corresponding hemisphere that contains all the points

forming, this normal represents a feasible parting direction for the die and punch without any interference at the mating surface during the stamping operation. Figure 24 and Figure 25 show surfaces with the surface normals and the unit spheres with the normals centered at their origin.

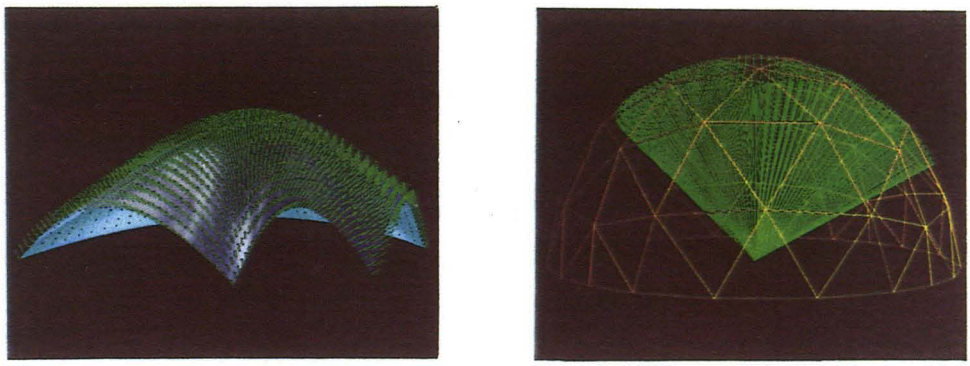


Figure 24 Hood shaped surface and the normals centered at the origin.

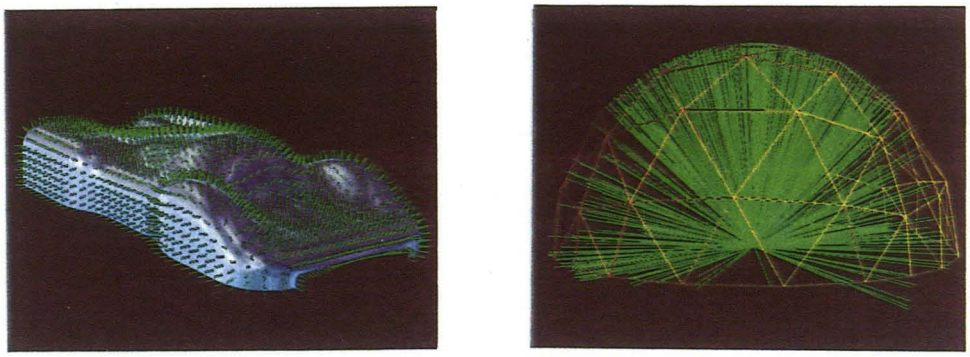


Figure 25 B-Spline surface of a car body and its Gaussian map

6.2 Determination of Geometric Strains

Using the result obtained from the constant area transformation explained in Chapter 3, it is possible to calculate a measure of resultant geometric strains on the surface using the theory of finite strain based on the homogenous deformation mode. This resulting strain distribution is yet another qualitative indicator useful to the surface and die designers.

6.2.1 Strain theory

As a solid is deformed, points in the body are displaced. Strain is defined in terms of such displacements but in a way as to exclude the effects of rigid body movements like pure translation or pure rotation. Consider a length l_0 which under loading changes to l such that $l \neq l_0$. Although, both translation and rotation may occur during the loading process, it is the change in length that is used to define strain as

$$e = \frac{l - l_0}{l_0} = \frac{\Delta l}{l_0} \tag{6.4}$$

where e is the nominal or engineering strain. For large strains, an alternative definition, proposed by Ludwik (1909), is more convenient. True strain, ε, also called logarithmic or natural strain, is defined such that every incremental length change is divided by the current length,

$$d\varepsilon = \frac{dl}{l} \tag{6.5}$$

which upon integration gives

$$\varepsilon = \ln \frac{l}{l_0} \tag{6.6}$$

True strains are more convenient than engineering strains because:

1. *True strains for equivalent deformation in tension and compression are identical except in sign.*
2. *True strains are additive, the total strain being the sum of the incremental strains. This is not true for engineering strains.*
3. *The volume change is related to the sum of the three normal strains, and volume constancy relationship is given by*

$$\varepsilon_x + \varepsilon_y + \varepsilon_z = 0 \tag{6.7}$$

4. *If the strains are small, then true and engineering strains are nearly equal.*

Expressing Equation (6.6) as

$$\varepsilon = \ln\left(\frac{l_0 + \Delta l}{l_0}\right) = \ln\left(1 + \frac{\Delta l}{l_0}\right) = \ln(1 + e) \tag{6.8}$$

a series expansion results in

$$\varepsilon = e - \frac{e^2}{2} + \frac{e^3}{3!} + \ldots\ldots \tag{6.9}$$

as $e \to 0, \varepsilon \to e$ (Hosford and Caddell, 1983.)

6.2.2 Finite strain

In metal deformation processes involving finite (plastic) strains, the material properties depend upon the current state of the strain which is itself a function of the deformation history. It is customary to attempt an integration over the strain path, but in general the integral cannot be evaluated explicitly. Furthermore, it is usually impractical to measure three dimensional strains, for instance those arising in a forging operation. In the sheet metal forming process, most strain measurements are based on grid markings on a free surface. If the grid is measured after a small increment in deformation, the strain increment is, in general easily determined. The mode of deformation will dictate how readily the strain increment in each successive deformation step can be evaluated from the distorted grid, and how easily the strain increments can be integrated. In many instances, however, it is either

too time consuming or difficult to make incremental measures and only measurements of the initial and final shape of the grid are made.

6.2.3 Circular grid analysis

Largely by convention, in sheet metal stamping it is assumed that one principal axis of plastic strain increment is normal to the material surface, with the other two principal directions lying in the plane of the sheet. The well known circular-grid analysis technique (Dinda et al., 1981) for determining strains in sheet metal pressings is based on this hypothesis. The method also assumes that a grid circle, etched or printed on an un-deformed blank, is transformed into an ellipse on the surface of the pressing. By measuring the major and minor diameter of the ellipse, the principal surface strains are determined. Although over a large portion of the surface, circles transform into approximate ellipses, not every element deforms in this manner. In general, the deformation path of an element in an industrial pressing is not known precisely. The degree to which the measurements of a grid circle reflect the actual strain is a matter of conjecture and actually depends upon the complexity of the strain path. In the presence of high strain gradients the grid-circle method is inadequate, since a grid circle will undergo severe distortion and no longer resembles an ellipse after deformation. The grid circle analysis technique however, reduces the strain determination to a two dimensional problem since the deformed ellipse lies on a relatively flat surface. Another major limitation of the circular grid analysis procedure is the physical limit of the circle size due to the equipment capability for etching or printing the circle on the blank. Ideally, if the circles are small enough, then even in the areas of high strain gradients, the grid circles will undergo only plane strain along the principal directions. In what follows, the deformation in two dimensions is considered and two important straining modes are discussed.

6.2.4 Homogenous deformation

The deformation process which transforms straight lines into straight lines and circles into ellipses is usually referred to as homogenous deformation (Love, 1944). When the

principal axes are fixed within each material element, the deformation is termed pure homogenous strain, which is exemplified by the pure shear mode. Only the principal axes remain fixed in direction, all other line elements rotate. It is well known that the final shape of a material element following a prescribed homogenous strain, can also be realized by imposing a pure homogenous mode followed by a rigid body rotation. This poses the question of equivalence of the strain paths. Sowerby et al. (1982) show that the pure homogenous deformation mode leads to a simple finite strain tensor, analogous to the infinitesimal strain tensor but without the need to introduce any simplifying assumptions.

Sowerby et al. (1982) proposed a method which eliminates some of the problems encountered in strain calculations using the circular grid analysis. Specifically, the uncertainty of assessing the major and minor axis of an ellipse is eliminated, and an improved averaging of the strain within an element becomes available. Emphasis is placed on the pure homogenous mode, since it leads to a simpler finite strain tensor. This method does not overcome certain problems encountered with the grid-analysis technique, such as the presence of strain gradients. However, the author believes that since this method is amenable to an interactive design format, it is possible to make the elemental surface entities small enough to predict stable results.

6.2.5 Two dimensional homogenous strain

If the stress components and the corresponding strain components on one reference plane vanish (e.g., z axis plane, z being a principal direction), a condition of biaxial or plane stress exists. To investigate how the normal and shear strains components vary with orientation in the x-y plane, a cut is made at some arbitrary angle, ϕ, as shown in Figure 26. The strains on this plane are e_ϕ and υ_ϕ.

From equilibrium considerations, it can be shown that

$$e_\phi = \frac{e_x + e_y}{2} + \frac{e_x - e_y}{2}\cos 2\phi + \gamma_{xy}\sin 2\phi \qquad (6.10)$$

and

$$\gamma_\phi = \frac{-(e_x - e_y)}{2}\sin 2\phi + \gamma_{xy}\cos 2\phi \qquad (6.11)$$

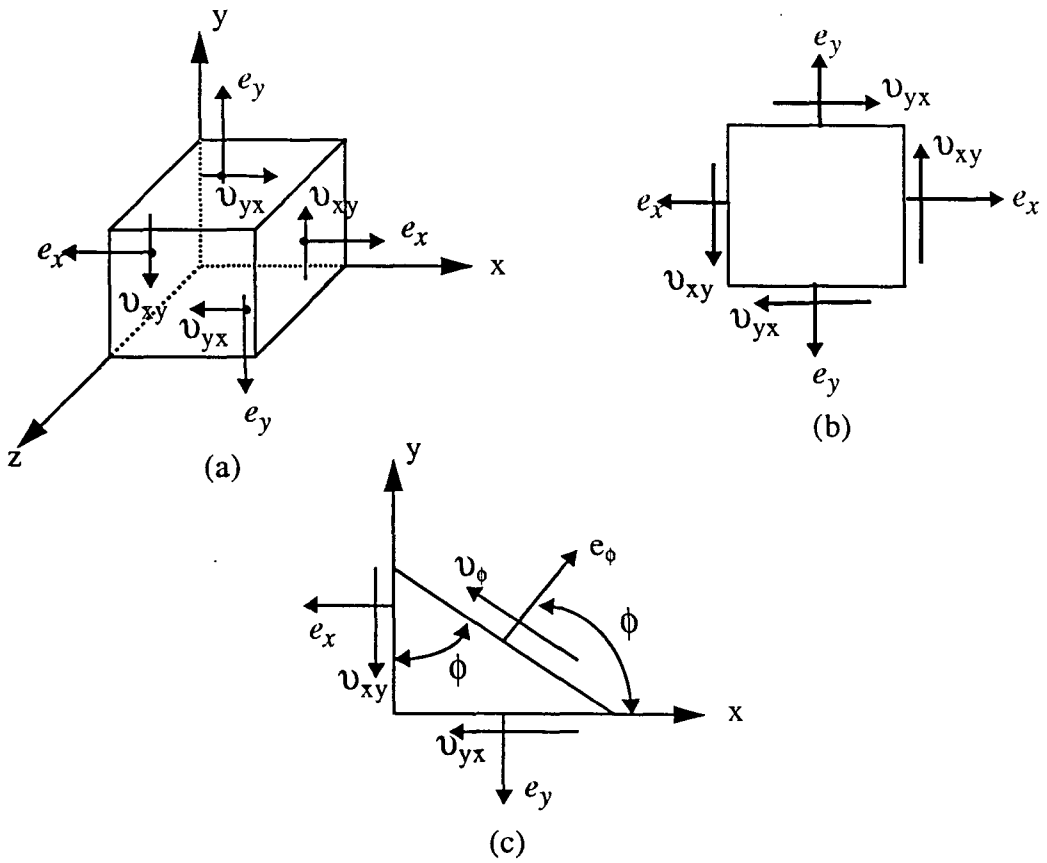


Figure 26 (a) Element in a state of biaxial (plane) strain. (b) Strains as viewed in the x-y plane. (c) A cut at an arbitrary angle, ϕ.

Values of e_ϕ on planes where υ_ϕ is zero are the two principal strains that lie in the x-y plane. Where υ_ϕ is zero, from Equation (6.11)

$$\tan 2\phi = \frac{2\gamma_{xy}}{e_x - e_y} \qquad (6.12)$$

and Figure 27 provides the corresponding graphical relation. Using the values of sin2ϕ and cos2ϕ from Figure 27, Equation (6.10) becomes

$$e_1, e_2 = \frac{1}{2}(e_x + e_y) \pm \frac{1}{2}[(e_x - e_y)^2 + 4\gamma_{xy}^2]^{1/2} \tag{6.13}$$

where e_1 and e_2 are the values of the two principal strains that act in the x-y plane. Using Figure 27, the largest shear stress in the x-y plane is

$$\gamma_{max} = \frac{1}{2}[(e_x - e_y)^2 + 4\gamma_{xy}^2]^{1/2} \tag{6.14}$$

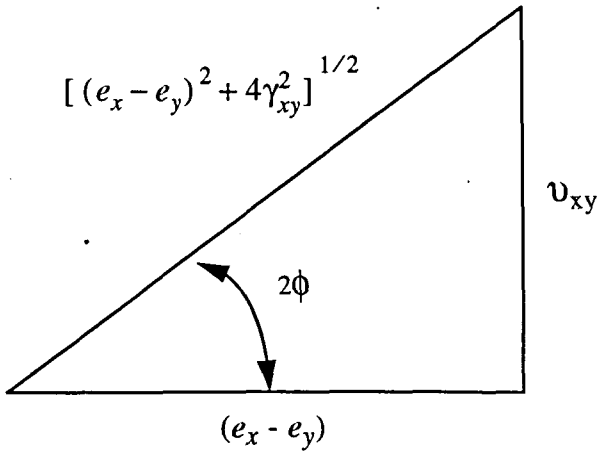


Figure 27 Relationship of resulting strains used to derive Equation (6.10) and Equation (6.11)

Homogenous deformation of a two dimensional element is illustrated in Figure 28. A small un-strained element ABCD is deformed into A'B'C'D' and the displacements are shown as F_{11}, F_{12}, F_{21} and F_{22}. To relate the initial and the deformed shapes, it is mathematically convenient to consider reference coordinate axes which are fixed in space. In Figure 28, the reference axes have been arbitrarily selected to be parallel to the sides of the un-

deformed grid. The new coordinates (x, y) of a vertex are a linear function of the initial coordinates, say (X_0,Y_0), and can be expressed as

$$\begin{aligned} x &= F_{11}X_0 + F_{12}Y_0 \\ y &= F_{21}X_0 + F_{22}Y_0 \end{aligned} \tag{6.15}$$

or

$$\bar{x} = \underset{\sim}{F} \cdot \bar{X}_0 \tag{6.16}$$

where $\bar{X}_0$ is a vector in the un-deformed configuration and is mapped into $\bar{x}$. Strains F_{11}

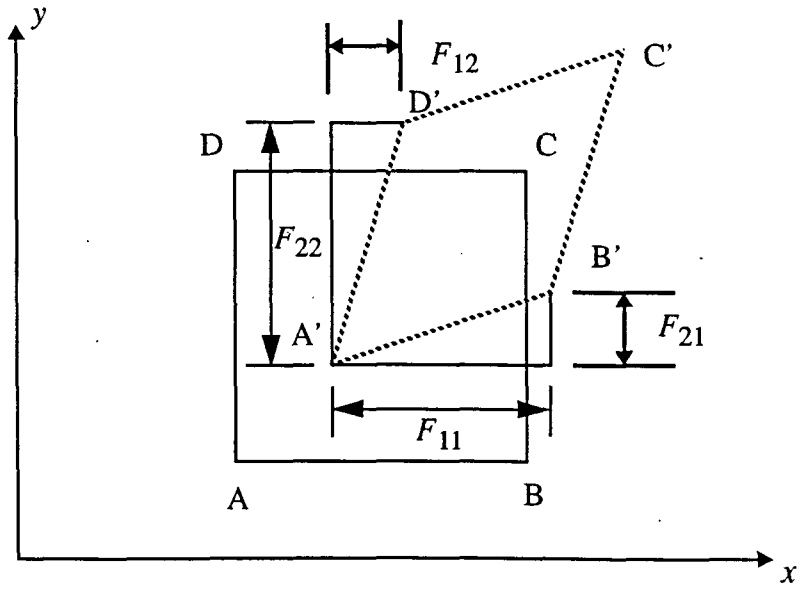


Figure 28 Homogenous deformation in two dimensions of a quadrilateral grid element

and F_{22} correspond to linear change and therefore are the principal strains. F_{12} and F_{21} are related to the angle change between the sides AB and AD of the element and therefore correspond to the shear strain. In general, $F_{12} \neq F_{21}$. The deformation gradient tensor, $\underset{\sim}{F}$ depends upon the basis selected, but the deformation tensor, $\underset{\sim}{C}$, where,

$$\underline{C} = \underline{F}^T \cdot \underline{F} \tag{6.17}$$

does not. When the deformation tensor is symmetric, the deformation mode is called pure homogenous transformation. The tensor $\underline{C}$ is symmetric and its eigen-values are usually referred to as the principal stretch values squared.

6.2.6 Pure homogenous deformation

Pure homogenous deformation is characterized by a symmetric tensor $\underline{F}$. The shear strain components are the equal, $F_{12} = F_{21}$ and F_{11} and F_{22} represent the principal engineering strains,i.e. $\frac{\text{final length}}{\text{initial length}}$, λ_1 and λ_2. The orientation of the principal axes is obtained from

$$\tan 2\theta = \frac{2F_{12}}{F_{11} - F_{22}} \tag{6.18}$$

The principal logarithmic surface strains, ε_1 and ε_2 are determined from

$$\varepsilon_{1,2} = ln\,(\lambda_{1,2}) \tag{6.19}$$

while the third principal strain, ε_3, is given by the incompressibility assumption

$$\varepsilon_1 + \varepsilon_2 + \varepsilon_3 = 0 \tag{6.20}$$

For the state of plane strain however, ε_3 is neglected. The representative or equivalent strain, $\bar{\varepsilon}$ is given by

$$\bar{\varepsilon} = \sqrt{\frac{2}{3}(\varepsilon_1^2 + \varepsilon_2^2 + \varepsilon_3^2)} \tag{6.21}$$

Figure 29 shows two arbitrary triangular elements. Let Figure 29.(b) represent a triangular element on the deformed surface, and Figure 29.(a) represent the same element af-

ter the mapping to the plane using the mapping strategy. An arbitrary orthogonal set of reference axes is constructed such that the vertex of the un-deformed triangle coincides with the origin of the axes. As discussed in the preceding section, the coordinates (x, y) of a vertex can be expressed as a linear function. Hence,

$$\begin{aligned} x &= F_{11}X + F_{12}Y \\ y &= F_{21}X + F_{22}Y \\ \bar{x} &= \underline{F} \cdot \bar{X} \end{aligned} \tag{6.22}$$

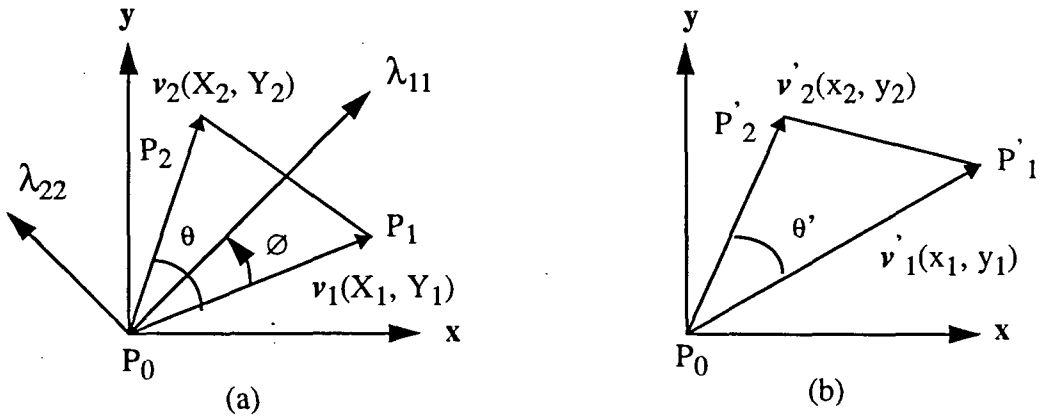


Figure 29 Pure homogenous deformation of a triangular element

Since the deformation is assumed to be homogenous, the four coefficients in (6.22) are constants over the region where the deformation occurs.

The coefficients can be evaluated from the vectors ν_1 and ν_2 before and after deformation, formed from the three points P_0, P_1, P_2 of Figure 29. All vectors are mapped on the same plane by employing rigid body rotations which will not change the magnitudes of the vectors. The coefficients of (6.22) can be determined with reference to Figure 29

$$
\begin{aligned}
x_1 &= F_{11}X_1 + F_{12}Y_1 \\
y_1 &= F_{11}X_1 + F_{12}Y_1 \\
x_2 &= F_{11}X_2 + F_{12}Y_2 \\
y_2 &= F_{11}X_2 + F_{12}Y_2
\end{aligned}
\tag{6.23}
$$

or, in matrix form, this equation yields

$$
\begin{Bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \end{Bmatrix} = \begin{bmatrix} X_1Y_1 & 0 & 0 \\ X_2Y_2 & 0 & 0 \\ 0 & 0 & X_1Y_1 \\ 0 & 0 & X_2Y_2 \end{bmatrix} \begin{Bmatrix} F_{11} \\ F_{12} \\ F_{21} \\ F_{22} \end{Bmatrix} \tag{6.24}
$$

The inversion of this matrix will result in the values of the four coefficients

$$
\begin{Bmatrix} F_{11} \\ F_{12} \\ F_{21} \\ F_{22} \end{Bmatrix} = D \begin{bmatrix} Y_2 & -Y_1 & 0 & 0 \\ -X_2 & X_1 & 0 & 0 \\ 0 & 0 & Y_2 & -Y_1 \\ 0 & 0 & -X_2 & X_1 \end{bmatrix} \begin{Bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \end{Bmatrix} \tag{6.25}
$$

where $D = \dfrac{1}{(X_1Y_2 - X_2Y_1)}$

Independent of the selected axes, the coefficients F_{11}, F_{12}, F_{21} and F_{22} can be combined in particular way to always yield the same components of a symmetric second order tensor, $\underline{C}$, known as Green's-deformation tensor such that

$$
\underline{C} = \underline{F}^T \cdot \underline{F} \tag{6.26}
$$

Upon expanding the above equation, the components of the symmetric tensor are given by

$$\begin{aligned} C_{11} &= F_{11}^2 + F_{21}^2 \\ C_{12} = C_{21} &= F_{11}F_{12} + F_{21}F_{22} \\ C_{22} &= F_{12}^2 + F_{22}^2 \end{aligned} \tag{6.27}$$

Hence the principal value of the elongation ratios squared, the orientation of the principal axes and the principal logarithmic strains in the initial referential configuration are given by

$$\begin{aligned} \lambda_{11}^2, \lambda_{22}^2 &= \frac{C_{11}+C_{22}}{2} \pm \sqrt{\left(\frac{C_{11}+C_{22}}{2}\right)^2 + C_{12}^2} \\ \tan 2\theta &= \frac{2C_{12}}{C_{11}-C_{22}} \\ \text{and} \quad \varepsilon_{11}, \varepsilon_{22} &= ln\,(\lambda_{11}, \lambda_{22}) \end{aligned} \tag{6.28}$$

This strain calculation procedure was implemented as another manufacturability function to complement the blank development algorithm described in Chapter 3. It provides the designer with even more information about the possible behavior of the surface under scrutiny. The results for two example surfaces used for blank development are shown in Table 2.

6.3 Discussion

Strain analysis of the two example surfaces are shown in Figure 30 and Figure 31. The strains are displayed as color coded images for emphasis and ease of perception. Strains are colored in the standard color spectrum range between violet and red signifying

pure tension and compression respectively. The principal major strains for the B-spline model of the B-Spline surface in Figure 31 shows large tension in the lower edge of the surface, predicting possible fracture during forming. The strain analysis for the hood shaped surface shows a rather uniform spread of surface strains as shown in Figure 30.

Table 2 Strain Analysis results

Strains	Major Strain		Minor Strain		Average Strain	
Objects	Maximum	Minimum	Maximum	Minimum	Maximum	Minimum
Bezier Surface	4.5722	3.4658	0.1451	-0.6928	3.7433	2.8298
B-Spline Surface	5.6082	0.1935	0.2691	-7.5837	6.6632	0.3397

6.4 Robustness Issues

Strain analysis procedure followed in the course of this research is based on the work done by Sowerby and Chakravarti (1983) and Sowerby et al. (1982). It is the opinion of the author that, the present strain model is too simplistic and is perhaps valid for small deformations only. Rigorous analysis and testing of the procedure needs to be conducted to arrive at a better methodology to determine the geometric strain.

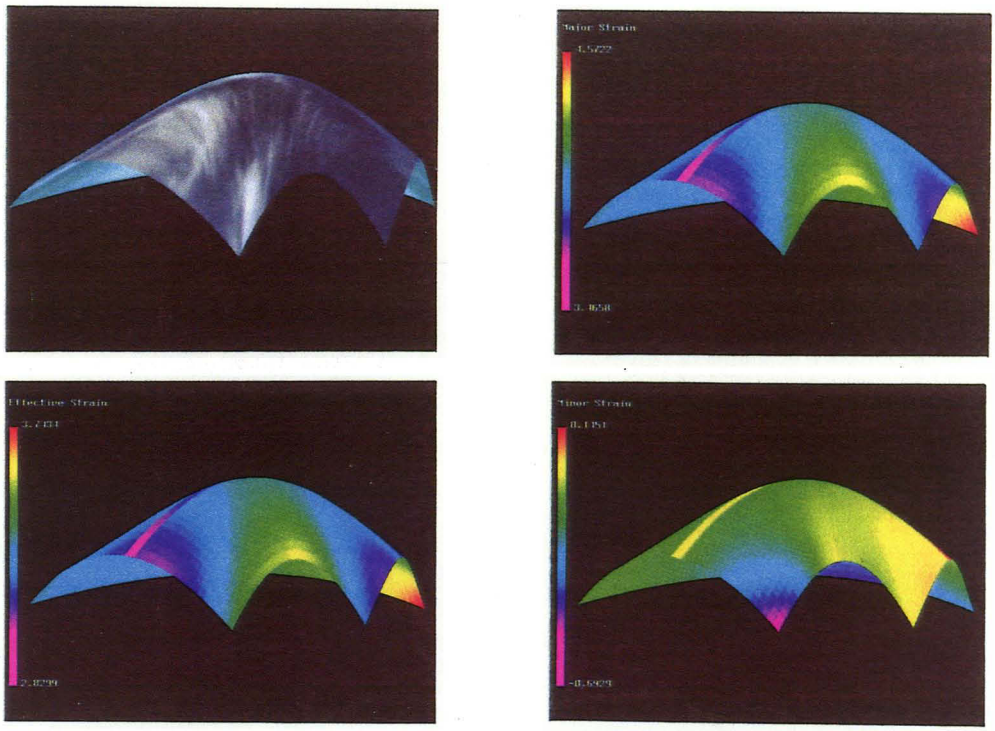


Figure 30 Strain analysis of Bezier surface.

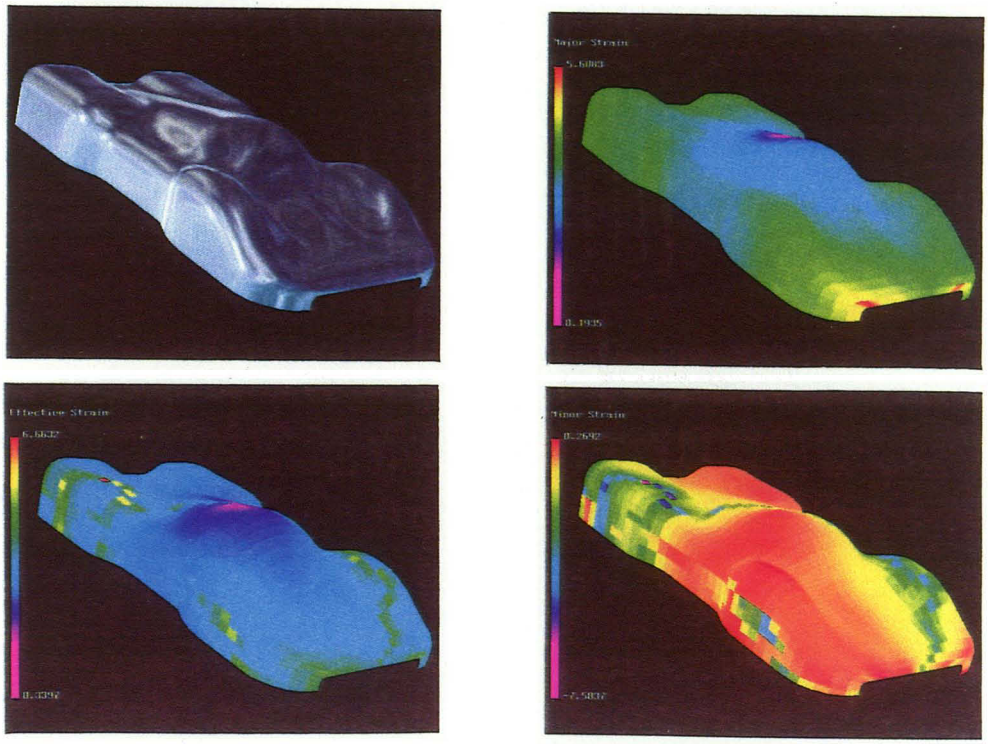


Figure 31 Strain analysis of B-spline surface.

CHAPTER 7
PRESS FORMING VISUALIZATION

Visualization of manufacturing processes is emerging as a powerful new tool for manufacturing assessment. The motivating philosophy behind this new technology is concurrent design - specifically, bringing the two aspects of manufacturing, namely, product design and manufacture to work in tandem for cost and quality effectiveness. Press forming dies form the basis of any sheet metal processing industry. It would be useful for a die maker or a sheet metal product designer to be able to visualize the deformation behavior of the sheet for a particular die shape. In this chapter, a virtual press forming environment is described which addresses this need. The input to the software is the surface definition as a NURBS surface in standard IGES format. The industry standards for the major components of the press forming system are applied to the input surface, and the system is configured accordingly. The designer can then interactively deform an initial blank through successive forming stages and visually examine the various components for a qualitative assessment of the process.

7.1 Introduction

Traditionally, in a manufacturing environment, product designers have had little experience and information about the manufacturing processes required to create products they design on their drawing boards. Statistical tables and manufacturing reference manu-

als and other textual forms of information are not quite effective in information dissemination to the designers, especially in a fast paced production environment. Visualization or animation gives the user a perspective of the product or process which cannot be gleaned readily from statical aids. The need for prototype manufacture for a new product or conducting a trial run for a process can be eliminated by using simulation and visualization tools to model the process or product in its entirety, thus drastically reducing the design cost of the product or process. Visualization also enables the user to view a process in intermediate stages at no extra cost which would not be possible when dealing with real-time physical systems. Analysis of present day manufacturing practice reveals that product and process design take up 75% of a product life cycle (Nevins and Whitney, 1989), the major portion of this time being spend on iterative design of all the product elements. Until recently, visualization techniques were limited due to unavailability of affordable hardware and software. Today manufacturing is completely technology driven. The advent of fast microprocessors and graphics dedicated technology to led to visualization emerging as an effective technique for the designer.

To augment the formability assessment algorithms presented in CHAPTER 3, a software tool is designed to aid the sheet metal surface and die designers during the preliminary design stage. The software creates a virtual press forming environment, showing the major components of the process which allows the designer to interactively watch the deformation process. The software utilizes the geometric properties of the surfaces and curves to represent and manipulate the objects. Visualization using force equilibrium relations and constitutive equations of the materials need extensive computation time and do not readily lend themselves to interactive performance. Moreover, the press forming deformation process has still not been well understood and at the present level of research in the field, generalized predictive models have not been sufficiently tested. In the spirit of the material property-free approach as explained in CHAPTER 3, the visualization software design phi-

losophy is based on purely geometric transformations and is broadly targeted as a visual aid for designers and for technical presentation purposes.

7.2 Press Forming

A sheet metal panel is usually produced in a double action press as shown in Figure 32. The press forming operation combines the action of bending and drawing of sheet metal. The tooling for the operation consists of two major assemblies - a matching punch and die which define the shape of the panel, and a draw ring or binder which surrounds the punch and die and controls the flow of the sheet inwards as it is formed to the required geometry. In the visualization module, only the first major assembly comprising the punch and die, guide posts, and punch and die holders are represented.

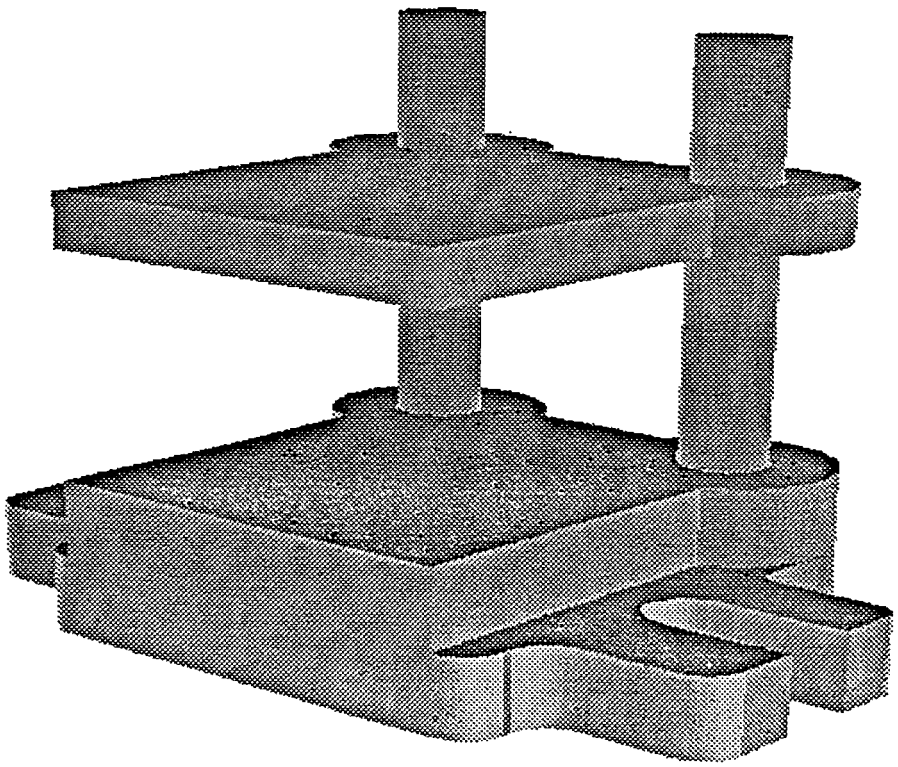


Figure 32 Schematic of a die and punch assembly

7.3 System Description

The input to the software is the final surface definition in the Interactive Graphics Exchange Specification (IGES) format as an IGES-128 entity. An IGES-128 entity is a Non Uniform Rational B-splines (NURBS) surface which is used as the basis for all geometric operations in the software. IGES is a widely accepted standard for exchange of geometric models among the various CAD systems. The visualization software accepts the IGES entity and converts it into an internal spline definition array for further processing. The bounding box dimensions of the surface are calculated and the information is used to calculate the press forming assembly for the surface. ASTME commercial standards are used for the determining the dimensions of the assembly (ASTME, 1975). The configuration employed in this software tool is a regular back post medium sized die set. The components used in the system are shown in Figure 33.

7.4 Die-set Terminology

7.4.1 Punch holder

The die-set top member is called the punch holder (or punch shoe). The punch holder is are used by the designer for squaring and locating the punch components of the die. The upper side bears against the underside of the punch ram and the punch is fastened to the lower finished surface. The great majority of standardized die-sets have the guide bushings mounted on the punch holder.

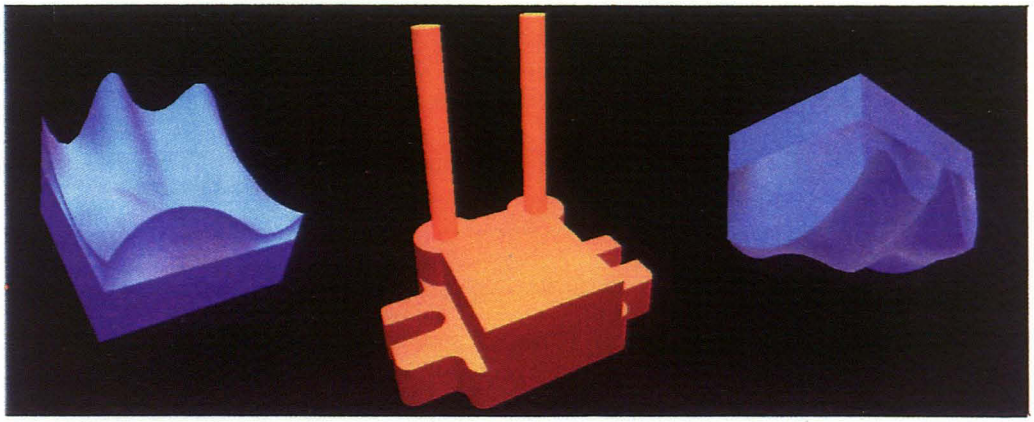
7.4.2 Die holder

This is the lower working member of the die set. Its shape corresponds to the punch holder except that it is provided with clamping flanges having slots for bolting the die holder to the bolster plate of the press. The die holder is made thicker than the punch to compensate for the weakening affect of slug and blank holes which are machined through it.

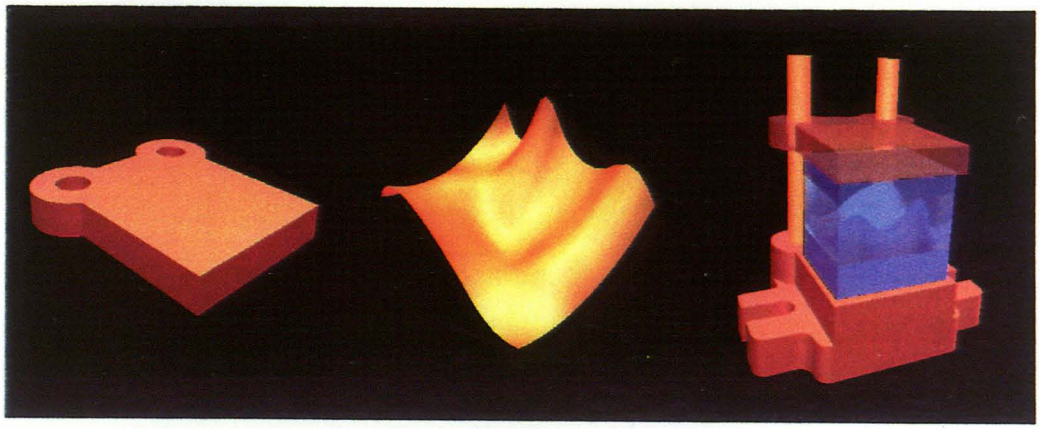
Standardized die-sets have guideposts mounted on the die holder. Standards employed for small and medium size dies are:

Punch holder thickness - 1 1/4 inches

Die holder thickness -1 1/2 inches



Die **Die holder** **Punch**



Punch holder **Surface** **Assembly**

Figure 33 Components of the punch and die assembly in the visualization system

7.4.3 Guide posts

Guide posts are precision ground pins which are press fitted into accurately bored holes in the die holder. They engage guide bushings to align punch and die components with a high degree of closeness and accuracy. Guide posts are specified at least 1/4 inch

shorter than the shut height of the die. For purposes of the project this standard has not been maintained.

7.4.4 Shut height of die

Shut height is the distance from the bottom surface of the die-holder to the top surface of the punch holder, excluding the shank, and measured when the punch is in the lowest working position. Overall height of the guide posts must always be an adequate amount less than the shut height in order to ensure that the ram will not strike against the ends of the guideposts. For cutting dies, if possible, the guideposts should be short enough to accommodate the total amount that the shut height will be lowered because of sharpening.

7.4.5 Die-set material specification

The following are the standard material specification for the die-set.

1. *Semi-steel - a gray iron or cast iron containing 10 to 25 percent steel.*
2. *Hot-rolled boiler plate - 1018 to 1026 SAE steel.*
3. *A combinations of platens, one a casting and the other steel.*
4. *Aluminium, magnesium or special alloys as well as soft, semi-hardened, or hardened tool steels.*

7.4.6 Improvements in die-set design

In the early days of mass produced stamped parts, tolerances and alignment requirements for dies making crude blanks and forms were not critical. The early die-set was an open set which had no built-in guidance or alignment means and thus depended on the accuracy of the press itself to maintain alignment. It was necessary either to have accurate presses for close tolerance work, or to compromise on the quality and the uniformity of the parts produced.

The first die-set to be self aligning was made with two cast or steel ground plates which were strapped together and through-bored for two or more pins. Pins were pressed in one plate and were guided through the holes in the second plate, which were lapped for a sliding fit. Later, bosses were cast on the upper plates (cast sets) or were welded (in the case of steel sets) to give added length to the bored holes, thereby improving alignment.

Sometimes oversize holes were made in the upper moving plate and the guide pins were inserted through a soft liner to help reduce friction on the guide pins and reduce scoring and galling, keeping the die-set accurate and free-working longer. The next improvement was the development of inserted bushing, usually pressed in with an interference fit and lapped to fit the guide pins. This, basically, is the die-set now in use. These improvements made the tooling alignment the job of the die-sets rather than the press, and more critical sizes and tolerances could be held on the parts produced.

The punch and die inner surface follow the contours of the surface to be formed. In actual forming operation, the die and punch are designed to allow for the blank spring back i.e., elastic reaction to press release, material tension and compression.

7.5 System Modeling

7.5.1 Surface

The surface is decoded from the input IGES format to an internal definition to be used in conjunction with *DT_NURBS* (Boeing Computer Services, 1990). The surface at regular intervals to generate the surface points. Normals required for lighting calculations in the graphic environment are calculated at each surface point. The surface is then rendered using triangular meshes.

7.5.2 Punch and die assembly:

Once the dimensions of the assembly are calculated from the initial surface configuration, the die set object calculation modules are called to configure the object dimensions. Each component of the assembly is broken down into primitives and then "stitched" together to form the composite model. The object surface points are "skinned" using triangular meshes and rendered. The flow chart for creation of the various entities is shown in Figure 34.

7.5.3 Hyperpatches

The visualization tool developed for this project is based on geometric modeling of the surface using tensor products of parametric curves. The dynamic process of forming is simulated by employing hyperpatches and showing incremental constant parameter surfaces of the resulting parametric solid to represent the various stages of deformation of the blank (Mortenson, 1985).

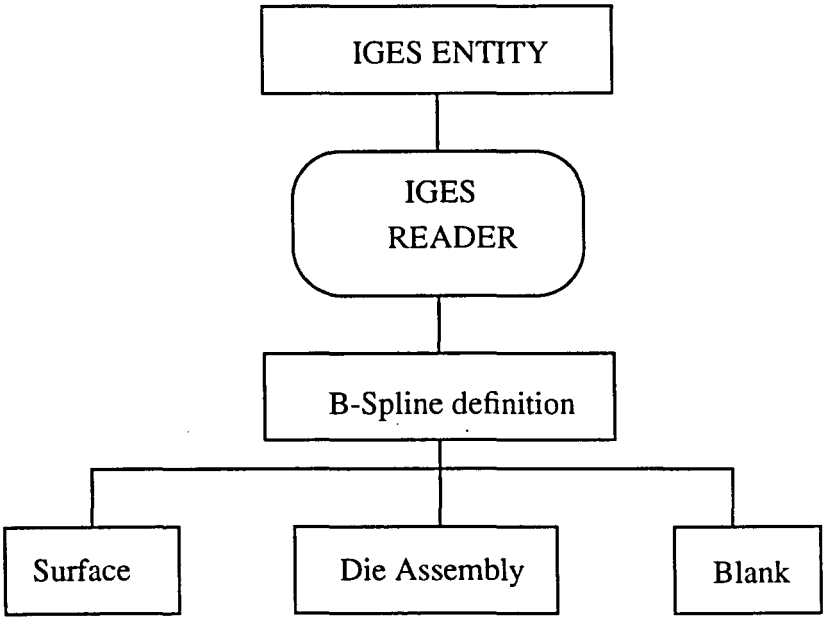


Figure 34 Flow chart for creation of geometric entities

A description of the B-spline representation scheme is presented in CHAPTER 3. A hyperpatch (Mortenson, 1985) is a patch-bounded collection of points whose coordinates are given by continuous, three-parameter, single-valued mathematical functions of the form: $x = x(u,v,w)$, $y = y(u,v,w)$, $z = z(u,v,w)$. The parametric representation of a tensor product surface in B-spline form is given by:

$$\mathbf{S}(u, v, w) = \sum_{i=0}^{m} \sum_{j=0}^{n} \sum_{r=0}^{t} \mathbf{P}_{ijr} N_{i,k}(u) N_{j,l}(v) N_{r,q}(w) \qquad u, v, w \in [0, 1]$$

where $S(u,v,w)$ is a three dimensional vector function of control points P_{ijr} arranged on an $(m+1) \times (n+1) \times (t+1)$ topologically cuboidal network and $N_{i,k}(u)$, $N_{j,l}(v)$ and $N_{r,q}(w)$are the degree k, l *and* q B-spline basis functions, respectively. Fixing the value of one of the parametric variables results in an isoparametric surface within, or on the boundary of, the hyperpatch in terms of the other two variables, which remain free.

Incremental isoparametric surfaces are used to simulate the deformation process. This choice is motivated by the ease of implementation and visually realistic transformation from the blank to the final surface. It must be noted that the technique used here is not validated for physically accuracy. To simulate the dynamic process of sheet metal forming, the bicubic B-spline surface is converted into a hyperpatch with a linear curve in the w direction as described in Appendix B. At $w = 0$, the isoparametric surface is the undeformed state of the blank, and the deformation is completed at $w = 1$ state.

A quadratic function is used to generate values of w (direction of punch travel). The isoparametric surface is generated, representing the partially deformed blank. The process is repeated until the final configuration of the surface is obtained, i.e., when $w = 1$. The details of the hyperpatch formulation is explained in Appendix B and the flow chart for the animation process is shown in Figure 35.

7.5.4 Graphics Environment

DT_NURBS spline geometry library (Boeing Computer Services, 1990) routines are used for evaluation of tensor product splines, calculation of normals and partial derivatives. The internal data structure for the isoparametric surface and the hyperpatch in *DT_NURBS* format are described in Appendix A.The user interface was developed using *Forms* 2.0 - a public domain software package for creating graphical user interfaces. The user interface includes features to change the object material attributes such as reflectance, color and transparency. The program has the capability of saving and retrieving a particular view of the scene. On a Silicon Graphics 4D/420VGXT workstation, the simulation runs in real time. The various sequence of events during the forming cycle are shown in Figure 36.

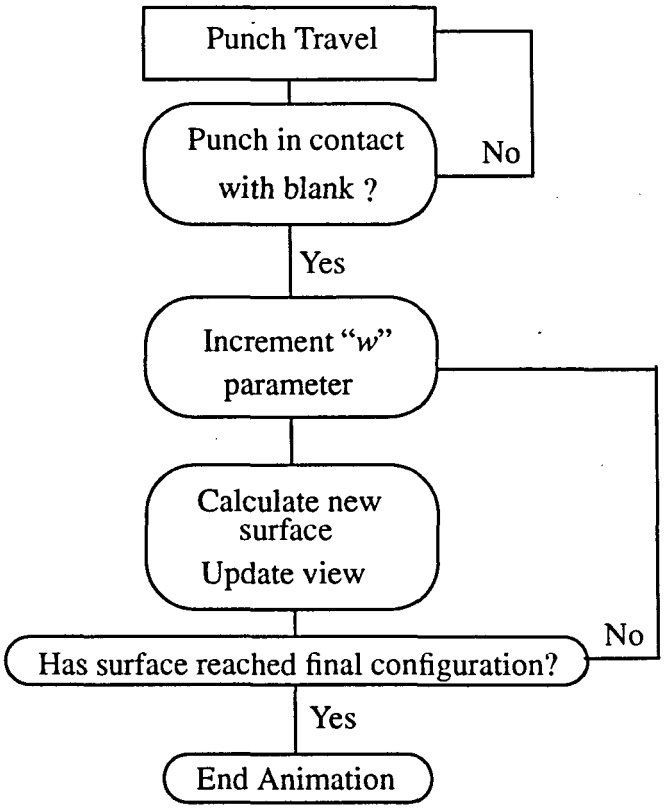


Figure 35 Flow chart showing the program structure of the forming process

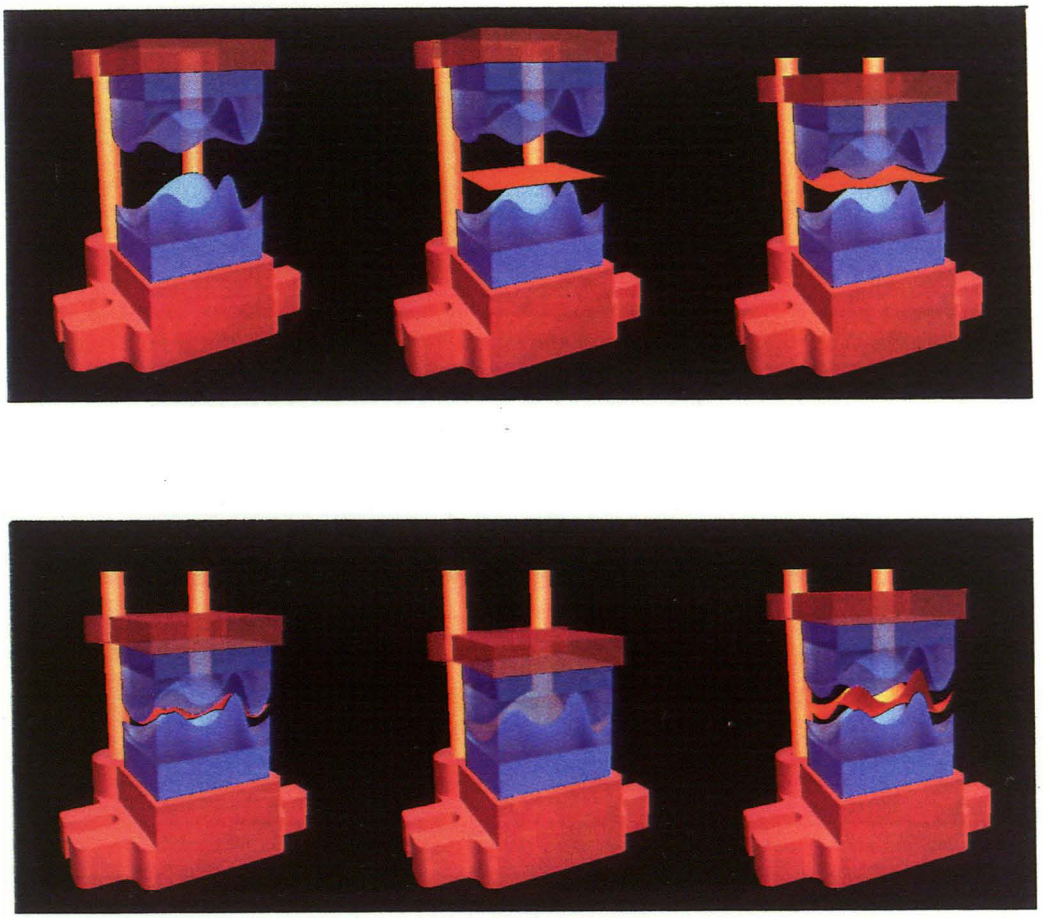


Figure 36 Pictures showing the various stages in the deformation process.

CHAPTER 8
CONCLUSIONS AND FUTURE RESEARCH

8.1 Conclusions

A general formability assessment tool for surfaces manufactured by press forming has been developed. An efficient set of algorithms for constant area transformation of an arbitrary shaped surface to a plane has been formulated. The direct application of the algorithms is in blank development of a design surface - a useful tool for both surface and die-designers. A set of geometric constraints based on visibility is developed and a methodology for calculating geometric strains is implemented. Finally, a press forming simulation tool is described for visualizing the forming process. Several examples are presented to substantiate the research.

The results indicate that this constant area transformation algorithm provides a robust and computationally efficient technique for press forming blank development. The efficiency of the technique is due to the fact that geometric feasibility is independent of material property. It is interesting to note that other researchers have reached similar conclusions in considering the manufacturability of parts comprised of layered composite materials (Tam and Gutowski, 1990; Gutowski et al., 1991). Thus, the simplicity of the underlying algorithm and its corresponding linear time complexity make this constant area technique quite suitable for implementation as an interactive design aide. The methodology for strain calculations provided approximate results. The results should be verified by actual experi-

mentation. The author is of the opinion however, that the methodology holds promise and needs to be analyzed rigorously during the course of future research.

8.2 Future Research

Research related to the constant are transformation is continuing on several fronts. In fact, the technique developed in the course of this Master's theses was initially conceived as an ancillary function for sculptured surface model synthesis (Oliver et al., 1993). Although the constant area transformation technique has emerged as a useful tool by itself, a major focus has been to develop a general formability constraint to be incorporated into the sculptured surface model synthesis technique (Oliver and Theruvakattil, 1993). As a stand alone design aide, the constant area transformation algorithm will be enhanced in several ways. For example, the method will be extended to accommodate more general surface models to handle merging of multiple fronts from several peaks and stages simultaneously. In addition, other mapping strategies, such as volume conservation and/or minimum energy path, will be investigated. The present method for strain evaluation can be improved to provide a more reliable measure for design specification. It is hoped that these additional capabilities will provide a more accurate assessment of formability for complex surfaces.

REFERENCES

ASTME, 1975, *Die Design Handbook*, McGraw-Hill, New York.

Barata da Rocha, A., Barlat, F., and Jalinier, J.M., 1985, "Forming Limit of Anisotropic Sheets in Complex Strain Path," *Proceedings of the symposium on Computer Modeling of Sheet Metal Forming Processes*, sponsored by The Metallurgical Society, held at the 12th annual Automotive Material Symposium, Ann Arbor, Michigan, pp. 93-106.

Boeing Computer Services, 1990, *David Taylor Research Center Spline Geometry Library*, Carderock Division, Naval Surface Warfare Center, Bethesda, Maryland.

Chen L.L., and Woo, T.C., 1992, "Computational Geometry on the Sphere with Application to Automated Machining," *ASME journal of Mechanical Design*, Vol. 114, pp. 288-295.

Chen L.L., Chou, S.Y., and Woo, T.C., 1992, "Separating and Intersecting Spherical Polygons: for Computing Machinability on 3-, 4-, and 5-axis Numerical Controlled Machines," *Submitted to ACM Transactions on Graphics*.

Chu, E., 1983, "New Horizons in Computer-Aided Design of Sheet Metal Stampings," *Ph.D. Dissertation*, McMaster University, Montreal.

Chu, E., Soper, D., Gloekl, H., and Gerdeen, J.C., 1985, "Computer-Aided Geometric Simulation of Sheet Metal Forming Processes," *Proceedings of the symposium on Computer Modeling of Sheet Metal Forming Processes*, sponsored by The Metallurgical Society, held at the 12th annual Automotive Material Symposium, Ann Arbor, Michigan, pp. 65-76.

Clements, J.C., 1981, "A Computer System to Derive Developable Hull Surfaces and Tables of Offsets," *Marine Technology*, Vol. 18, No. 3, pp. 227-233.

Clements, J.C., and Leon,L.J., 1987, "A Fast, Accurate Algorithm for the Isometric Mapping of a Developable Surface," *SIAM Journal for Mathematical Analysis*, Vol. 18, No. 4, pp. 966-971.

Dieter, G.E., 1961, *Mechanical Metallurgy*, McGraw-Hill Book Company, New York.

Dinda, S., James, K.F., Keeler, S.P., and Stine, P.A., 1981, *How to use Circle Grid Analysis for Die Tryout*, American Society of Metals, Metal Park, Ohio.

Goodwin, G.M., 1968, "Application of Strain Analysis to Sheet Metal Forming Problems in the Press Shop," *Society of Automobile Engineers*, Paper No. 680093.

Graham, R.L., 1972, "An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set," *Info. Proc. Lett.* 1, pp. 132-133.

Gutowski, T.G., Hoult, D., Dillon, G., and Gonzalez-Zugasti, J., 1991, "Differential Geometry and the Forming of Aligned Fibre Composites," *Composites Manufacturing*, Vol. 2, No. 3, pp. 147-152.

Higashi, M., Mori, T., Taniguchi, H., and Yoshimi, J., 1985, "Geometric Modeling for Efficient Evaluation of Press Forming Severity," *Proceedings of the symposium on Computer Modeling of Sheet Metal Forming Processes*, sponsored by The Metallurgical Society, held at the 12th annual Automotive Material Symposium, Ann Arbor, Michigan, pp. 21-36.

Hill, R., 1950, *The Mathematical Theory of Plasticity*, Oxford University Press, Oxford.

Hoffman, C.M., 1989, *Geometric & Solid Modeling*, Morgan Kaufmann Publishers, Inc., San Mateo, California.

Hosford, W.F., and Cadell, R.M., 1983, *Metal Forming, Mechanics and Metallurgy*, Prentice-Hall Inc., Englewood Cliffs, New Jersey.

Keeler, S.P., 1965, "Determination of Forming Limits in Automotive Stampings," *Society of Automotive Engineers*, Paper No. 650535.

Kokkonen, V., 1985, "Modeling of Forming Processes for Tool Design and Manufacturing at Volvo," *Proceedings of the symposium on Computer Modeling of Sheet Metal Forming Processes*, sponsored by The Metallurgical Society, held at the 12th annual Automotive Material Symposium, Ann Arbor, Michigan, pp. 13-20.

Lee, C.H., and Kobayashi, S., 1973, "New Solutions to Rigid-Plastic Deformation Problems using a Matrix Method," *ASME Journal of Engineering for Industry*, Vol. 95, pp. 865-873.

Lee, E.H., and Tehrani, A.A., 1986, "The structure of Constitutive Equations for Finite Deformation of elastic-plastic materials with strain-induced anisotropy," *Proceedings of the NUMIFORM'86 conference*, Gothenburg, pp.29-36.

Light, R.A., and Gossard, D.C., 1982, "Modification of Geometric Models through Variational Geometry," *Computer Aided Design*, Vol. 14, No. 4, pp. 209-214.

Lin, V.C., Gossard, D.C., and Light, R.A., 1981, "Variational Geometry in Computer-Aided Design," *Computer Graphics*, Vol. 15, No. 3, pp. 171-177.

Love, A.E.H., 1944, *A Treatise on the Mathematical Theory of Elasticity*, Dover Publications, New York.

Ludwick, P., 1909, *Elemente der Technologischen Mechanik*, Springer, Berlin.

Makinouchi, A., 1986, "Finite Element Modeling of Draw-bending process of Sheet Metal," *Proceedings of the NUMIFORM'86 conference*, Gothenburg, pp.327-332.

Meggido, N., 1983, "Linear Time Algorithm for Linear Programming in R^3 and Related Problems," *SIAM Journal of Computing*, Vol. 12, No. 4, pp. 759-776.

Mortenson, M.E., 1985, *Geometric Modeling*, Wiley, New York.

Nagpal, V., Subramanium, T.L., and Altan, T., 1979, AFML-TR-79-4168, *AFWAL Material Laboratory*, Battelle.

Nevins, J.L. and Whitney, D.E., 1989, *Concurrent Design of Products and Processes*, McGraw Hill, New York.

Oliver, J.H., and Theruvakattil, P.C., 1993, "Sculptured Surface Model Based on Functional Design Constraints," *To be presented at the ASME Design Automation Conference*, Albuquerque, New Mexico, September, 1993.

Oliver, J.H., Theruvakattil, P.C., Nair, N.K., 1993, "Towards Automated Generation of Sculptured Surface Models," *Proceedings of the 1993 NSF Design and Manufacturing Systems Conference*, Vol. 1, pp. 663-640.

Onate, E., and Saracibar, C.E., 1988, "Finite Element Analysis of Sheet Metal Forming Problems using a Viscous Voided Shell Formulation," *Proceedings of the Euromesh 233 Colloquium*, Sophia Antipolis, France, pp.163-178.

O'Rourke, J., Chien, C., Olson, T., and Naddor, D., 1982, "A new Linear Algorithm for Intersecting Convex Polygons," *Computer Graphics and Image Processing*, Vol. 19, pp. 384-391.

Piegl, L. and Tiller, W., 1987, "Curves and Surface Construction Using Rational B-Splines," *Computer-Aided Design*, Vol. 19, No. 9, pp. 485-497.

Redont, P., 1989, "Representation and Deformation of Developable Surfaces," *Computer-Aided Design*, Vol. 21, No. 1, pp. 13-20.

Sachs, G., 1935, "New Researches on the drawing of cylindrical shells," *Proceedings of the Institution of Automobile Engineers*, Vol. 9, pp.588-600.

Shamos, M.I., 1978, "Computational Geometry," *Ph.D Dissertation*, Department of Computer Science, Yale University, New Haven.

Shimada, T., and Tada, Y., 1989, "Development of Curved Surfaces using Finite Element Method," *Proceedings of the first international conference on Computer-Aided Optimum Design of Structures*, Recent Advances, Springer-Verlag, New York, pp. 23-30.

Shimada, T., and Tada, Y., 1991, "Approximate Transformation of Arbitrary Curved Surface into a Plane using Dynamic Programming," *Computer-Aided Design*, Vol. 23, No. 2, pp. 153-159.

Shin, S.Y., 1986, "Visibility in the Plane and its Related Problems," *Ph.D. Dissertation*, Department of Industrial and Operational Engineering, University of Michigan, Ann Arbor.

Sowerby, R., and Chakravarti, P.C., 1983, "The Determination of the Equivalent Strain in Finite, Homogenous Deformation Processes," *Journal of Strain Analysis*, Vol. 18, No. 2, 119-123.

Sowerby, R., Chu, E., and Duncan, J.L., 1982, "Determination of Large Strains in Metal Forming," *Journal of Strain Analysis*, Vol. 17, No. 2, pp. 95-101.

Takahashi, M., Okamoto, I., Hiramatsu, T., and Yamada, N., 1985, "Evaluation Methods of Press Forming Severity in CAD applications," *Proceedings of the symposium on Computer Modeling of Sheet Metal Forming Processes*, sponsored by The Metallurgical Society, held at the 12th annual Automotive Material Symposium, Ann Arbor, Michigan, pp. 37-50.

Tam, A.S., and Gutowski, T.G., 1990, "The Kinematics for Forming Ideal Aligned Fibre Composites into Complex Shapes," *Composites Manufacturing*, Vol. 1, No. 4, pp. 219-228.

Tang, S.C., Chu, E., and Samanta, S.K., 1982, "Finite ELement Prediction of Deformed Shape of an Automotive Body Panel during Preformed Stage," *Proceedings of the International Conference on Numerical Methods in Industrial Forming Processes*, Swansea, U.K, pp. 629-640.

Vegter, H., 1988, "A Finite Difference Model as a Basis for Developing New Constitutive Equations for the sheet deformation process," *Proceedings of the Euromesh 233 Colloquium*, Sophia Antipolis, France, pp.111-121.

Wang, N.M., 1970, "Large Plastic Deformation os a Circular Sheet caused by Punch Stretching," ASME Journal of Applied Mechanics, Vol. 37, pp. 431-440.

Wang, N.M., and Tang, S.C., 1986, "Analysis of Bending Effects in Sheet Metal Forming Operations," *Proceedings of the NUMIFORM'86 conference*, Gothenburg, pp.71-76.

Wagoner, R.H., 1985, "Constitutive Equations for Sheet Metal Forming Analysis," *Proceedings of the symposium on Computer Modeling of Sheet Metal Forming Processes*, sponsored by The Metallurgical Society, held at the 12th annual Automotive Material Symposium, Ann Arbor, Michigan, pp. 77-92.

Wagoner, R.H., Nakamachi, E., and Germain, Y., 1988, "Experience with Explicit and Implicit Finite Element Programs for Sheet Forming Analysis," *Proceedings of the Euromesh 233 Colloquium*, Sophia Antipolis, France, pp.131-138.

Zienkiwicz, O.C., 1984, "Flow Formulation for Numerical Solution of Forming Processes", *Numerical Analysis of Forming Processes*, John Wiley & Sons, U.K.

ACKNOWLEDGMENTS

This research was supported by the National Science Foundation (grant number DDM-9111122), the Office of Naval Research (grant number N0014-92-J-4092) and the Iowa Center for Emerging Manufacturing Technology. I am grateful for this support.

I thank my parents, granny, Nidhi and Nipun back in India for having patiently accepted my thesis as an excuse for not writing letters frequently. I like to thank my advisor Dr. Jim Oliver who has been a good friend and guide to me all through my student life at ISU. I wish to thank Professor Lin-Lin Chen and Dr. Shou-Yan Chou for their constant encouragement and technical assistance.

I finally thank Mom, Dad and my Libby family, friends Samir, Mannu, Jonathan, Stacy, Jawad, Sue and a pretty dalmatian called Heidi for being a part of an impressionable period of my life in the United States.

APPENDIX A

DATA STRUCTURE FOR TENSOR PRODUCT SPLINES

DT_NURBS routines for tensor product spline computations require the spline definition to be represented in a single dimension double precision array denoted by C and referred to as the C array. The iges reader converts the spline from the standard IGES format to the single dimension data structure. The data structure is described below.

Index Range	Name	Description
General parameters		
1	n	Number of parametric variables
2	m	Number of dimension variables
3 to n+2	k_j	Order of splines in each of n independent variables
n+3 to 2n+2	Nj	Number of B-spline coefficients for each of the n variables
2n+3 to 3n+2	*jspan*	index of last span located in each of the independent variables
Knots		
p_1 + 1 to p_2	z_1	knots in the 1st parametric direction
.	.	.
.	.	.
p_n+1 to p_{n+1}	z_n	knots in the n^{th} parametric direction
Coefficients		
q_1 +1 to q_2	$a_{j1,..jn}$	coefficients for the 1st independent variable
.	.	.
.	.	.
q_m + 1 to q_{m+1}	$a_{j1..jn}$	coefficients for the m^{th} independent variable

If n is 2 the result is a B-Spline surface. If n is 3 the result is a hyperpatch, in both the cases, m being 3 for the x, y, z coordinates. *jspan* is the index of the knot interval in which the last evaluation point was found. It is used primarily to speed up calculations. The indices p_i and q_i are computed as:

$$p_1 = 3n+2$$
$$p_i = p_{i-1} + C(2+i) + C(n + 2 + i)$$

and

$$q_1 = p_{n+1}$$
$$q_j = q_{j-i} + r$$

where r is the product C(n+3).....C(2n + 2).

APPENDIX B

BLANK DEVELOPMENT

Blank development involves creating a hyperpatch from a given B-Spline surface definition. This is done by calculating the bounding box coordinates of the surface and creating a planar surface coincident with the *XY* plane of the bounding box at its minimum *Z* value. The remaining steps are as follows.

1.Linearly interpolate between the four corners of the bounding box plane identified above to get a set of control points for the w direction of the hyperpatch.

2. Make the following modifications to the data structure for the hyperpatch.

$n = 3$

$k_3 = 3$

3. Add the knots for the *w* direction. The curve is the w direction is assumed to be quadric, so there are no internal knots, and the end knots are repeated.

4. Add the B-spline coefficients for the *w* direction

5. Now this array is send to the DT_NURBS routines for computations.