A B-SPLINE SURFACE MODELER WITH INTERACTIVE MODIFICATION OF THE CONTROL MESH

by

CARL ANDREW KEMNER

B.S. Kansas State University, 1984

A MASTERS THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Mechanical Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1987

Approved by:

Major Professor

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I would like to thank Professor J. Garth Thompson and the Department of Mechanical Engineering for making this research possible.

I would also like to thank my wife Jennifer and my daughter Bridget for enduring the extended duration of time needed to complete this project.

LIST OF TABLES AND FIGURES

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

Modeling of Free Form Surfaces

The modeling of complex three-dimensional surfaces used in the manufacture of cars, boats, airplanes and other designs requiring freeform surfaces has traditionally been done using solid model mock-ups. Mock-ups or "hard models" are very costly to build and difficult or impossible to iteratively modify.

"Soft models" are computer data sets that can be used to mathematically generate freeform curves and surfaces. In general there are two types of representations used for soft models: explicit functional forms and implicit functional forms. Explicit functional forms are based on combinations of explicit function in which the curves and surfaces are generated by directly varying the geometric parameters. Implicit functional forms are based on parametric representations of the explicit variables. Parametric variables are varied to generate the curves and surfaces.

A parametric representation consists of a set of vectors or control points and a functional form, usually a polynomial, in the parameter. The control points serve as weights to the terms in the functional form.

The first parametric representations used to represent curves and surfaces were Bezier curves and surfaces. This representation is based on a Bernstein polynomial basis and is a global approximation scheme. The number of control points is the same as the order of the curve and changing any one of the control points affects the shape of the entire curve.

A more general extension of the Bernstein basis is the B-spline basis, a local approximation scheme which can have an unlimited number of control points. Only a limited number of control points, equal to the order of the curve, have an effect on a given point. If the number of control points is equal to the order of the curve, the B-spline curve reduces to the Bezier form. There are two types of B-spline curves and surfaces that are commonly used in computer aided design (CAD), integral and rational. The integral B-spline is a non-homogenous vector valued function whose components are parametric polynomials. The rational B-spline is a homogenous vector valued function whose components are parametric polynomials.

The first part of this work will deal with the mathematical basis for the B-spline function. Next, the application of the B-spline approximation scheme to the representation of geometric models will be shown. Finally, a discussion of the program used to generate, modify, and display B-spline surfaces will be presented.

Implementations of the Present Work

The surface modeling features that have been implemented in this present work are: algorithms to generate B-spline surface streamlines and normals, a primitive display file, an interactive environment to globally change the data set for viewing, and an interactive environment to locally modify the data set to change the shape of the surface.

The B-spline algorithms implemented are those described by deBoor (ref 1) for the evaluation of the B-spline approximation and its derivatives. The extension of the algorithms to generate curves and surfaces are similar to those described by Riesenfeld (ref 2). The display file includes the B-spline control mesh, streamlines in both surface directions, and the normals at the intersection of each of the streamlines. The environment to globally change the data set for viewing permits the user: 1) to display the data in an isometric, 2) rotate the data about the Cartesian axes, or to scale the data. The environment to locally change the data is based on two ideas: first, the local support properties of the B-spline basis, and second, the use of a surface cursor rather than a screen cursor to indicate position in space.

In the next section the mathematical basis for the B-spline basis function and the form of the B-spline approximation scheme will be presented.

II. B-SPLINE APPROXIMATION: DEFINITIONS, FUNCTION, PROPERTIES, AND EVALUATIONS

Curves and surfaces are generally expressed as functions of the x, y, z, Cartesian coordinates. These functions can be either explicit:

$$f(x,y,z) = 0.0$$

or parametric

$$f(x)=x(t) \; ; \; f(y)=y(t) \; ; \; f(z)= z(t).$$

These functions are characterized by the degree or order (degree+1) of the variables. The set of all functions of the form

$$P_n = a_0 + a_1 t_1 + a_2 t_2^2 + \ldots + a_n t^n$$

is called the polynomials. The polynomials form a linear space in the mathematical sense. They are closed under the operations of addition and scalar multiplication. A basis for the polynomial space will be a set of functions whose linear combination can be used to represent all the members of that space. There are many different bases that can be used to represent the polynomial space. One class of functions that can be used as a basis for the polynomials is called splines. For a set of functions to be a spline, they must satisfy the following definition from Riesenfeld (ref 2).

Definition: Let $T= (t_0,t_1,\ldots,t_k)$ be a vector of reals such that $t_i < t_{i+1}$. A function N is called a (polynomial) spline function of degree K-1 (order K) if it satisfies the following two conditions:

1. N(K,T) is a polynomial of degree K-1 on each subinterval (t_i, t_{i+1}).

2. N(K,T) and its derivatives of order 1,2,...,K-2 are everywhere continuous, that is N is $C^{(K-2)}$.

The points t_i are called knots and T is called the knot vector. The B-spline functions form a basis for the polynomials, and the B-spline approximation uses this basis as its functional form. The B-spline approximation will be used in the geometric applications to represent curves and surfaces.

Defining the B-spline approximation's functional form.

The B-spline approximation has the following form:

$F(t)=\text{sumover}(i:0,n)\ A_i * N_{i,K}(t)$

1. F(t) : is the parametric function

2. A_i : is an ordered set of constants

3. $N_{i,K}(t)$: are the B-spline basis functions N(K,T)

4. K : is the order of the function (degree+1)

5. T : $(t_0, t_1, \ldots, t_k)$ is the knot vector

6. n : is the number of constants (zero indexed)

Details of the B-spline approximations

The use of the B-spline approximation will require the evaluation of the function and its derivatives with respect to the parameter. To evaluate the B-spline approximation requires: 1) the order of the basis, 2) the basis functions, 3) the knot vector, and 4) the constants A_i.

The order of curve is equal to the degree+1 or the number of coefficients in the associated K-1 degree polynomial.

The B-spline basis functions are defined by the following recursion formula (ref 1)

$$N_{i,K}(t) = \frac{t - t_i}{t_{i+K-1} - t_i} N_{i,K-1}(t) + \frac{t_{i+K} - t}{t_{i+K} - t_{i+1}} N_{i+1,K-1}(t)$$

$$N_{i,1}(t) = \begin{cases} 1.0 & \text{if } t_i <= t < t_{i+1}. \\ \text{else } 0.0 \end{cases}$$

The evaluation of the basis functions through order 4 is given in appendix A and plots of the conical basis functions through order 4 are shown in Figure 2-1. The basis functions are non-negative and the sum of all $N_{i,K}(t)$ for any value of t is equal to 1.0 (ref. 2). These properties are responsible for convexity of the curves and surfaces in geometric applications. The definition given leads to a one-sided basis function evaluation, but this does not effect the continuity at the point $t=t_{i+1}$. The number of basis function for a periodic B-spline is

$$\{N_{i,K}\} \ (i:0,m)$$

where m is the range of the knot vector without multiplicity.

The number of basis functions for a non-periodic B-spline is

$\{N_{i,K}\}$ (i:0,K+m-1)

where m is the same as for the periodic case.

The knot vector is a sequence of real numbers such that $t_i <= t_{i+1}$. If they are integers, the knot vector is uniform, otherwise the knot vector is considered non-uniform. A uniform knot vector can be used without any loss of generality in the definition of the function (ref 2). The B-spline basis can be periodic or non-periodic. If it is periodic the knot vector has the form

T:(0,1,2,...m+1)

and the basis functions are shown in Figure 2-2. If the knot vector is non-periodic then the form is

$$T: (\underbrace{0,0,\ldots,0}_{K},1,2,\ldots,m-1,\underbrace{t_{max},t_{max},\ldots,t_{max}}_{K})$$

The basis functions are shown in figure 2-3. The knot vector may have multiplicity up to K, that is $t_i=t_{i+1}=\ldots=t_{i+K-1}$. The effect of multiplicity is to reduce the differentiability of the basis functions by one degree for each multiple of t_i as can be seen in Figure 2-4, or similarly, it reduces the order of the curve on the span.

The constants can be scalar valued or r dimensional vectors. In graphics applications the constants are usually coordinate vectors with the function F(t) being the point on a parametric curve or surface. The characteristic that permits the efficient evaluation of the B-spline is the local nature of the basis function. That is, a maximum of K basis functions will be non-zero for any value of t. In computing F(t), the summation need only extend over the K non-zero basis functions $N_{i,K}(t)$ (i:j-(K-1),j) or

$$F(t)=\text{sumover}(i:j-(K-1),j)\ A_i * N_{i,K}(t) \quad j \text{ s.t. } t_j <= t < t_{j+1}.^{(*)}$$

The evaluation of the parametric derivatives require a reduced set of constants and the lower order basis functions of the span. The first derivative is evaluated in the following manner

$$F^{(1)}(t) = (K-1)\ \text{sumover}(i:0,n))\ A_i^{(1)} * N_{i,K-1}(t)$$

with $A_i^{(1)} = (A_i - A_{i-1}) / (t_{i+K-1} - t_i)$.

In general the derivatives are evaluated as follows

$$F^{(d)}(t) = (K-1)*(K-2)*...*(K-d)\ \text{sumover}(i:0,n)\ A_i^{(d)} * N_{i,(K-d)}(t)$$

and the constants are

$$A_i^{(0)} = A_i$$

$$A_i^{(d)} = (A_i^{(d-1)} - A_{i-1}^{(d-1)}) / (t_{i+K-d} - t_i) \quad :d > 0$$

with only i = j-(K-d)+1...j of $N_{i,(K-d)}(t)$ being non-zero and

(*) For a particular t, j is selected such that (s.t.) $t_j <= t < t_{j+1}$ and the range of the summation is j-(K-1) (K terms).

$$F^{(d)}(t) = (K-1)*...*(K-d)* \text{sumover}(i:j-(K-d)+1,j)\ A_i^{(d)} * N_{i,(K-d)}(t)$$

with j s.t. $t_j <= t < t_{j+1}$.

The evaluation of the derivative exhibits the same properties as the evaluation of the basis functions.

Properties of B-spline functions

B-spline functions exhibit some unique properties that are useful in geometric applications such as 1) local support, 2) (K-2) continuity and 3) they are variation diminishing. The property of local support is the result of the fact that only K basis functions are non-zero for any value of the parameter. The (K-2) continuity is guaranteed from the definition of the general spline function. The variation diminishing property (ref 2) is described as the ability of the approximation to reproduce linear functions exactly and to have no more zeros than the primitive itself.

Evaluations of the B-spline Functions

Algorithms for the efficient evaluation of the B-spline and its derivatives based on the basis function definition were given by deBoor (ref 1). The algorithm evaluates all the $N_{i,K}(t)$ that are not zero for a given t such that $t_i <= t < t_{i+1}$. These are shown in the table 2.1. The terms in the $(K-d)^{th}$ column of this table are the basis functions needed to evaluate the d^{th} derivative of the function F(t) at t.

$N_{i,1}(t)$	$N_{i-1,2}(t)$	......	$N_{i-K+2,K-1}(t)$	$N_{i-K+1,K}(t)$
	$N_{i,2}(t)$	......	$N_{i-K+3,K-1}(t)$	$N_{i-K+2,K}(t)$
		.	.	.
		.	.	.
		.	.	.
			$N_{i,K-1}(t)$	$N_{i-1,K}(t)$
				$N_{i,K}(t)$

Table 2-1. Required basis function evaluations

To generate the entries in this table column by column you must first define the following

$$N(r,s)=N_{i+r-s,s}(t)$$

$$DP(r)= t_{i+r} - t$$

$$DM(r) = t - t_{i+1-r} \quad r = 1,...,K$$

and the needed table entries are N(r,s) r = 1,...,s s = 1,...,K with N(r,s) = 0. for r > s or r < 1.

Substitute these into the recursion formula

$$N(r,s+1) = DM(s+1-r+1) * \frac{N(r-1,s)}{DP(r-1) + DM(s+1-r+1)}$$

$$+ DP(r) \frac{N(r,s)}{DP(r) + DM(s+1-r)}.$$

The algorithm to generate the N(r,s) entries in table 2-1 is given in Table 2-2 from deBoor (ref 1).

Set N(1,1) = 1.0

for s = 1, K-1 do:

set DP(s)= t_{i+s} - t, DM(s) = t - t_{i+1-s}

set N(1,s+1)=0.

for r = 1,...,s do:

set M=N(r,s)/(DP(r) + DM(s+1-r))

set N(r,s+1) = N(r,s+1) + DP(r) * M

set N(r+1,s+1) = DM(s+1-r) * M

Table 2-2. Algorithm to evaluate required basis functions.

Having calculated all the required basis functions, the evaluation of the B-spline function and its derivatives requires the summation of the appropriate products of derivative constants and basis function.

The one-sided definition of the basis function generates an exception in the evaluation of the non-periodic function when $t = t_{max}$. At that point all the $N_{i,K}(t)$ = 0. except $N_{n,K}(t)$ which is equal to 1.0, and the function is equal to the n^{th} constant at the point.

In the next section, the B-spline approximation will be applied to the modeling of freeform curves and surfaces.

a. order equal 1.

b. order equal 2.

c. order equal 3.

d. order equal 4.

Figure 2-1 Conical basis functions order 1 thru 4.

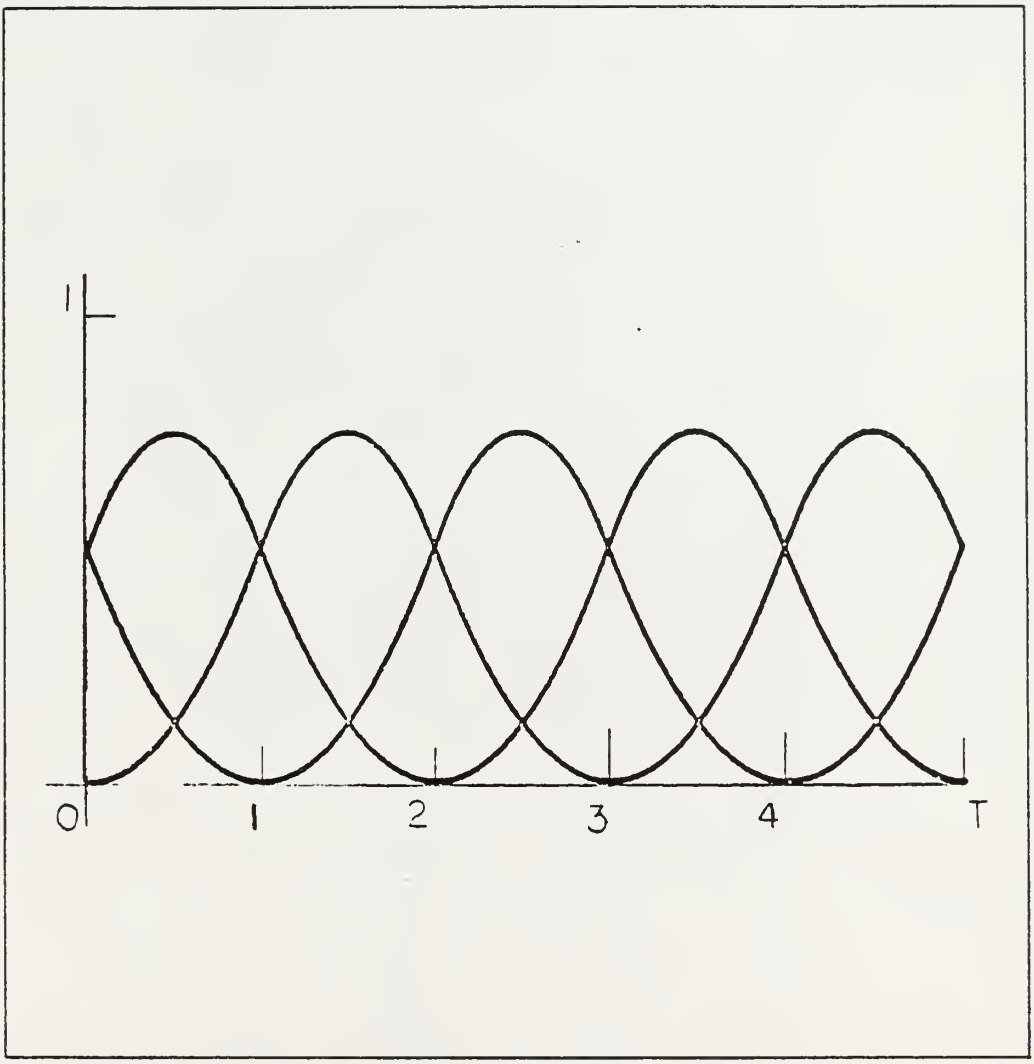


Figure 2-2 Periodic basis functions of order 3.

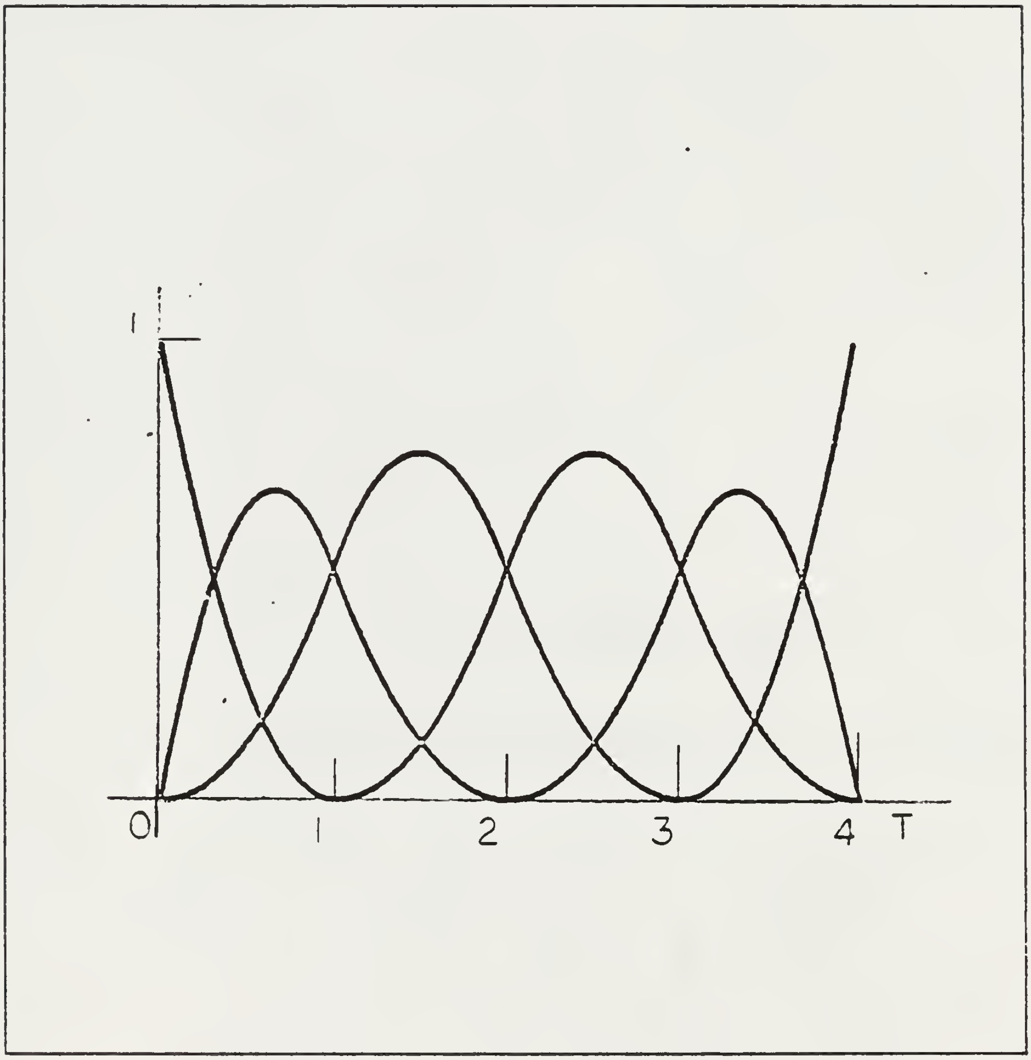


Figure 2-3   Non-periodic basis functions of order 3.

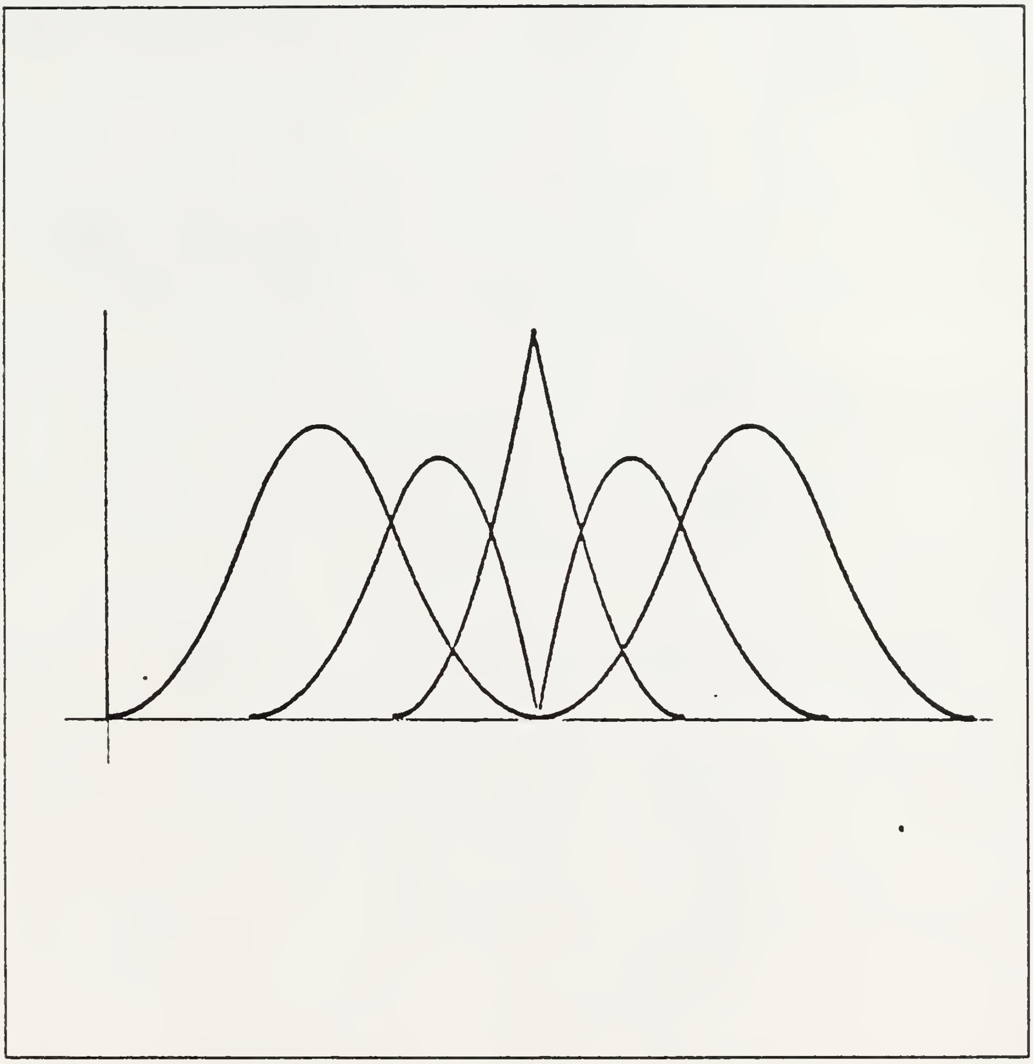


Figure 2-4 Non-periodic basis functions of order 3 with multiplicity.

III. GEOMETRIC APPLICATIONS OF B-SPLINE APPROXIMATION TO COMPUTER GENERATED CURVES AND SURFACES

The geometric application of the B-spline approximation is based on the use of an ordered set of vector-valued control points for the A_i's. These points represent a control polygon for curves (CP_i) or a control mesh ($CM_{i,j}$) for surfaces that will approximate the shape in the geometric coordinate system.

Types of B-spline Representations

Some of the ways that geometrical applications of B-splines can be categorized are by, 1) dimensionality, 2) periodicity, or 3) spatiality. Dimensionally, there are two forms of B-spline representations, the integral and the rational.

The original development of the integral form was done by Riesenfeld (ref 1) in 1973 and is a natural extension of the B-spline approximation to two or three Cartesian coordinates. In his thesis he reviews the mathematical basis for the B-spline approximation and shows that it is a proper extension of the Bernstein polynomial basis used by Bezier.

The original development of the rational form was done by Versprille (ref 2) in 1975, in which a homogenous representation is used to obtain desired geometric properties. The rational form permits the exact representation of conic sections. Since conics are the basis of most engineering drawing and since the integral form of the B-spline is a subset of the rational form obtained by setting the homogenous coordinate equal to 1.0, the rational form is better suited to geometric applications in engineering design. The development that follows will be for the rational form.

Geometric periodicity usually indicates whether the curve or surface is open, having distinct end points, or closed, and not having distinct endpoints. Open B-spline curves and surfaces have non-periodic knot vectors with distinct beginning and ending sequences of multiplicity K, and the control mesh or polygon is usually open with distinct starting and ending points. Closed B-spline curves and surfaces usually have periodic knot vectors, and the control mesh or polygon is closed on itself.

Spatiality refers to the type of geometric entity that is generated. B-spline curves have only one parametric variable and are evaluated by summing the product of the basis functions and the active control polygon. Surfaces have two parametric variables and are evaluated using a Cartesian product of basis functions in two directions and the elements of the active control mesh. The integral form of the B-spline curve is

$P(t)=sumover(i:j-(K-1),j)\ CP_i * N_{i,K}(t)$

with j such that $t_j <= t < t_{j+1}$.

Where P(t) and CP_i are vectors in the geometric space.

The rational form of the B-spline curve is:

$Q(t) = sumover(i:j-(K-1),j)\ CPQ_i * N_{i,K}(t)$

with j such that $t_j <= t < t_{j+1}$.

Where Q(t) and CPQ_i are vectors in the homogenous space and $Q_h(t)$ is the homogenous component of Q(t). Then

$$\begin{vmatrix} P(t) \\ 1.0 \end{vmatrix} = \begin{vmatrix} Q(t) \end{vmatrix} / Q_h(t)$$

B-spline curves can be applied in either two or three dimensional space.

The integral form of a B-spline surface is

$$S(u,v) = sumover(i:g-(I-1),g)\ sumover(j:h-(J-1),h)\ CM_{i,j} * N_{i,I}(u) * N_{j,J}(v)$$

with g such that $u_g <= u < u_{g+1}$ and h such that $v_h <= v < v_{h+1}$

where S(u,v) and $CM_{i,j}$ are vectors in the three dimensional geometric space. The rational form for a surface is

$$R(u,v) = sumover(i:g-(I-1),g)\ sumover(j:h-(J-1),h)\ CMR_{i,j} * N_{i,I}(u) * N_{j,J}(v)$$

with g such that $u_g <= u < u_{g+1}$ and h such that $v_h <= v < v_{h+1}$.

Where R(u,v) and $CMR_{i,j}$ are vectors in the homogenous space and $R_h(u,v)$ is the homogenous component of R(u,v). Then

$$\begin{bmatrix} S(u,v) \\ 1.0 \end{bmatrix} = \begin{bmatrix} R(u,v) \end{bmatrix} / R_h(u,v)$$

Geometric Properties of B-splines

Both integral and rational forms of the B-spline function exhibit similar geometric properties. Since the integral form is a subset of rational form, the development will focus on the rational form.

Both curves and surfaces in the B-spline representation are invariant under homogenous transformations, (rotations, scaling, translation, and perspective transformations) see (ref 5).

There are three different mechanisms that can be used to control the shape of the B-spline curves and surfaces corresponding to different parts of the functional definition. They are the order of the curve, the knot vector and the control points. The relationships between these mechanisms is not unique, and no intuitive scheme is known.

The order of the basis function determines the minimum number of control points required, the overall shape, and the continuity (K-2) of the resulting curve or surface.

The knot vector provides a mechanism that can be used to change the shape of curves and surfaces, particularly by using multiplicity to locally reduce the order of the basis functions. The knot vector for a non-periodic B-spline

$$T : (t_0, t_1, \ldots t_{n+K})$$

has n+K+1 elements which satisfy the relationship $t_i <= t_{i+1}$. Also, $t_0 = t_1 = \ldots = t_{K-1}$ and $t_{n+1} = t_{n+2} = \ldots t_{n+K}$ and for uniform B-splines with no multiplicities t_{max} = n-K+2. A common form for the knot vector is

$$T : (\underbrace{0,0,\ldots,0}_{K},1,2, \ldots \underbrace{t_{max},\ldots,t_{max},t_{max}}_{K})$$

The periodic knot vector is

$$T: (t_0, t_1, \ldots t_i \ldots, t_{n+1}) \text{ with } t_i = (i - K/2)_{mod\ n+1}.$$

The K/2 is integer division and the knot vector has no multiplicity.

The points in the control polygon or control mesh provide the most intuitive control over the shape of the curve or surface. There is a one to one correspondence between the number of control points and the number of K^{th} order basis functions. Two interesting special cases result from this relationship; one is associated with multiplicity of control points and the other is associated with collinear control points. If K control points CP_i are the same

$$CP_i = CP_{i+1} = \ldots = CP_{i+(K-1)}$$

then, since the sum of basis functions is always equal to 1.0, the curve must pass through that control point. If K control points are collinear, then the curve will be locally linear on that span. If

$$CP_i, CP_{i+1}, \ldots, CP_{i+(K-1)}$$

lie on some line, then the curve spanned by those control points will be linear. These special cases are the result of two properties of the B-spline function. The first, is variation diminishing, and second, it has local support. These two properties combined give rise to the property described later as the convex hull.

The geometric relationships between these control mechanisms are not unique In Rogers and Adams (ref 5) a method is described that attempts to directly relate repeated control points to multiplicities in the knot vector; in Tiller's article (ref 6), there are multiplicities in the knot vector without multiple control points. In general the knot vector must be specified such that the $N_{i,K}(t)$ are correctly defined with respect to the definition of the B-spline approximation.

B-spline Curves

Geometric properties of interest in the application of curves to engineering design are: 1) local support, 2) convex hull, 3) local linearity, and 4) numerical evaluations. Local support means that there will be at most K control points required to evaluate any point t on the curve. These control points are referred to as the active control points or the supporting polygon. The convex hull property states that any point on the curve will lie within the convex hull of the supporting polyhedron. Figure 3-1 illustrates the convex hull property for a planar curve. Local linearity implies that within any K^{th} order curve there can be locally linear portions if there are K collinear control points. Figure 3-2 illustrates local linearity.

Two vectors associated with geometric properties of interest are: the position vector and the tangent vector. The position vector is obtained for the rational B-spline by dividing by the homogenous component.

The formula for the evaluation of points on a B-spline curve was given at the beginning of this section. The evaluation of the tangent vector to the B-spline curve is the vector extension of the derivative of the B-spline approximation derivative, with the derivative constants becoming a set of derivative control points evaluated in vector form for d=1.

To evaluate the rational B-spline tangent vector:

first evaluate the parametric derivatives in homogenous form

$$Q^{(1)}(t) = \text{sumover}(i{:}j-(K-1)+1, j)\ CPQ_i^{(1)} * N_{i,K-1}(t)$$

j s.t. $t_j <= t < t_{j+1}$.

Then the geometric tangent vectors are

$$\begin{bmatrix} P^{(1)}(t) \\ 0.0 \end{bmatrix} = \left[Q_h(t)*Q^{(1)}(t) - Q(t)*Q_h^{(1)}(t) \right] / Q_h^{\,2}(t).$$

The B-spline approximation is defined as being left continuous for the interval $t_i <= t < t_{i+1}$. For <u>open</u> curves, as the parameter approaches the maximum value of the knot vector t_{max}, all the basis functions approach zero except $N_{n,K}(t)$ which approaches 1.0. The curve is equal to the end point.

B-spline Surfaces

The geometric properties of interest in application of B-spline surfaces to engineering design are similar to those for curves. These are: local support, convex hull, locally planar surfaces and numerical evaluations. For a surface there are two parametric variables u and v. Each of the parameters is associated with a basis function which has an order, a knot vector, and a control mesh instead of a polygon.

Local support for a surface means that for any point (u,v) on the surface there will be a control mesh that will be at most I by J points needed to evaluate it and its derivatives. I is the order in the u direction and J is the order in the v direction. The mesh will be called the active mesh or supporting polyhedron.

The convex hull for a surface is defined as the space bounded by the supporting polyhedron, and a point on the surface will lie within that space. Figure 3-4 shows the convex hull property for a surface. Portions of a surface can be made to be locally linear by making I control points, in the u direction or J control points, in the v direction collinear. Portions of the surface can be made to be locally planar by making an I*J array of control points coplanar.

The two properties required for surface applications are the position vector and the unit normal to the surface at the point. Points on the B-spline surface are evaluated using a Cartesian product of the active control mesh and the appropriate basis functions in each direction.

This was shown at the beginning of this section for both the rational and integral form.

Unit normals to the surface are evaluated by taking the cross product of the geometric tangent vectors in the u and v directions. The tangent vectors can be evaluated using the parametric partial derivatives of R(u,v). $R^{(u:1)}$ is the partial derivative of R(u,v) with respect to u, and $R^{(v:1)}$ is the partial derivative of R(u,v) with respect to v. R(u,v), $R^{(u:1)}$, $R^{(v:1)}$ are homogenous vectors. To evaluate the partials the homogeneous derivative control meshes, $CMR_{i,j}^{(u:1)}$ and $CMR_{i,j}^{(v:1)}$, are used. The tangent vectors, $S^{(u:1)}$ and $S^{(v:1)}$ and the normal vector, NV, are geometric vectors. The procedure to evaluate the unit normal to the surface is:

1. Evaluate the parametric partial derivatives of R(u,v).

   a. evaluate the derivative control points in each direction

   $$CMR_{i,j}^{(u,v)} = CMR_{i,j}$$

   for the partial of R(u,v) with respect to u

   $$CMR_{i,j}^{(u,1)} = CMR_{i,j}^{(u,v)} - CMR_{i-1,j}^{(u,v)} / (u_{i+I-1} - u_i)$$

   (i:g-(I-1)+1,g) (j:h-(J-1),h)

   and for the partial of R(u,v) with respect to v

   $$CMR_{i,j}^{(v,1)} = CMR_{i,j}^{(u,v)} - CMR_{i,j-1}^{(u,v)} / (v_{i+J-1} - v_i)$$

   (i:g-(I-1),g) (j:h-(J-1)+1,h).

b. evaluate the Cartesian product of derivative control points and the reduced order basis functions for each partial.

$$R^{(u:1)} = \text{sumover}(i:g-(I-1)+1,g)\,\text{sumover}(j:h-(J-1),h)\; CMR^{(u,1)}_{i,j} * N_{i,I-1}(u) * N_{j,J}(v)$$

$$R^{(v:1)} = \text{sumover}(i:g-(I-1),g)\,\text{sumover}(j:h-(J-1)+1,h)\; CMR^{(u,1)}_{i,j} * N_{i,I-1}(u) * N_{j,J}(v)$$

2. Calculate the geometric tangent vectors $S^{(u:1)}$ and $S^{(v:1)}$.

$$\begin{bmatrix} S^{(u:1)} \\ 0.0 \end{bmatrix} = \left[R^{(u:1)} * R_h - R * R_h^{(u:1)} \right] / R_h^{2}$$

$$\begin{bmatrix} S^{(v:1)} \\ 0.0 \end{bmatrix} = \left[R^{(v:1)} * R_h - R * R_h^{(v:1)} \right] / R_h^{2}$$

3. Evaluate the cross product to obtain the geometric normal to the surface.

$$\text{Normal Vector} = NV = S^{(u:1)} \times S^{(v:1)} = \begin{bmatrix} i & j & k \\ S_x^{(u:1)} & S_y^{(u:1)} & S_z^{(u:1)} \\ S_x^{(v:1)} & S_y^{(v:1)} & S_z^{(v:1)} \end{bmatrix}$$

$$= (NV_x, NV_y, NV_z)$$

4. Normalize the normal vector to get the unit normal to the surface at R(u,v).

$$\text{Unit Normal Vector} = NV / \text{SQRT}(NV_x^2 + NV_y^2 + NV_z^2).$$

For open surfaces two special boundary cases exist. First, when either of the parameters is equal to the maximum knot value in that direction, the supporting mesh degenerates to a supporting polygon along the edge in the other direction. Figure 3-4 shows this case. Second, when both parameters are equal to the maximum knot values, the supporting mesh degenerates to the corner control point.

This completes the discussion of the geometric application of the B-spline approximation. The next section will describe the program that was used to display and modify B-spline surfaces.

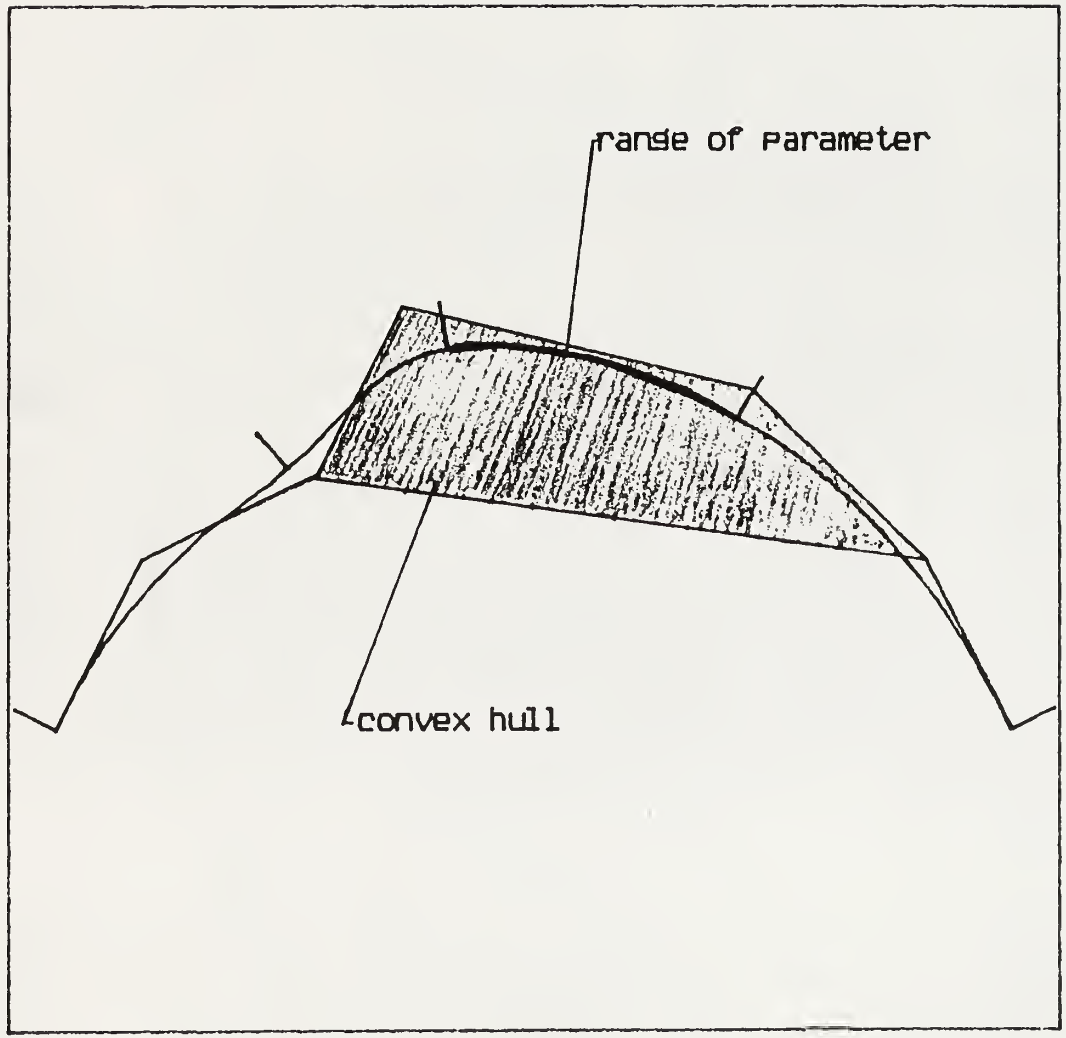


Figure 3-1 Convex hull property for a curve.

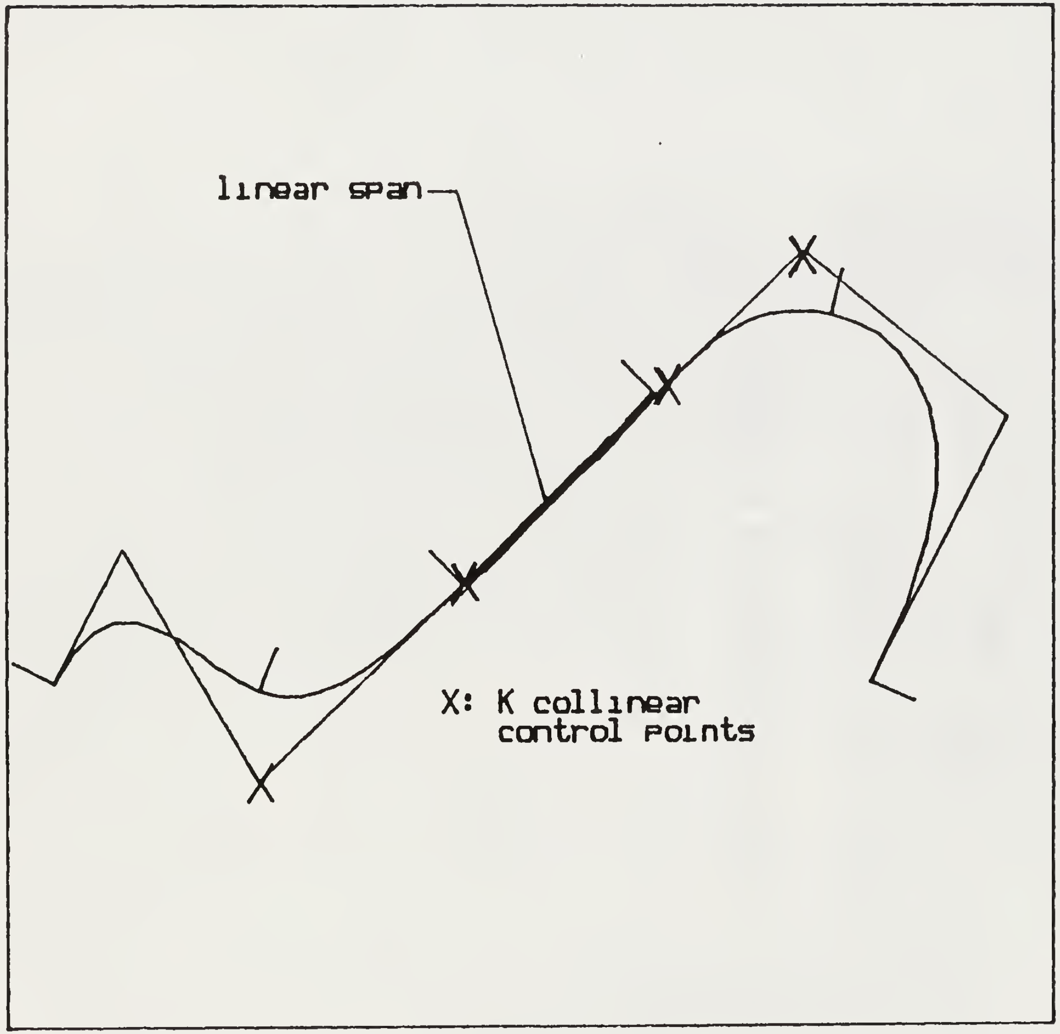


Figure 3-2 Local linearity for a curve.

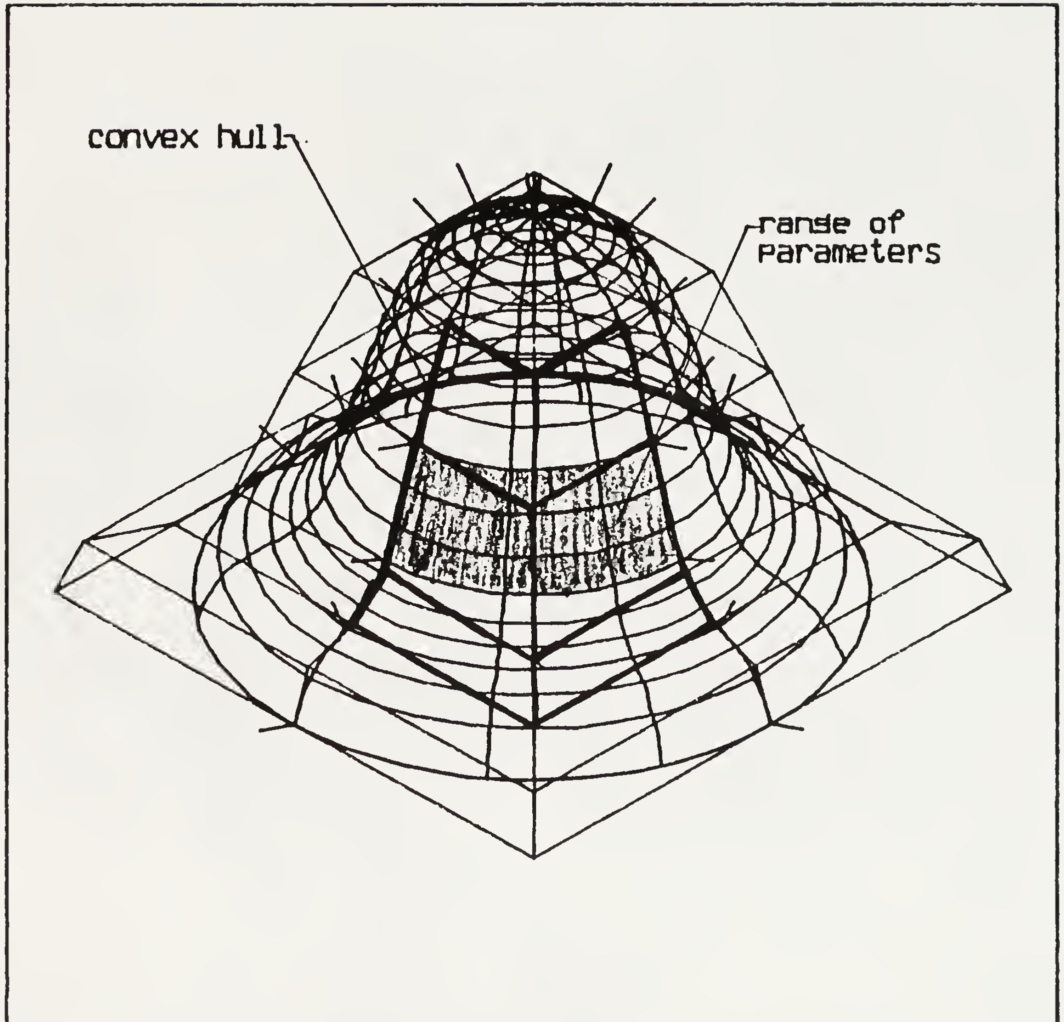


Figure 3-3   Convex hull for a surface.

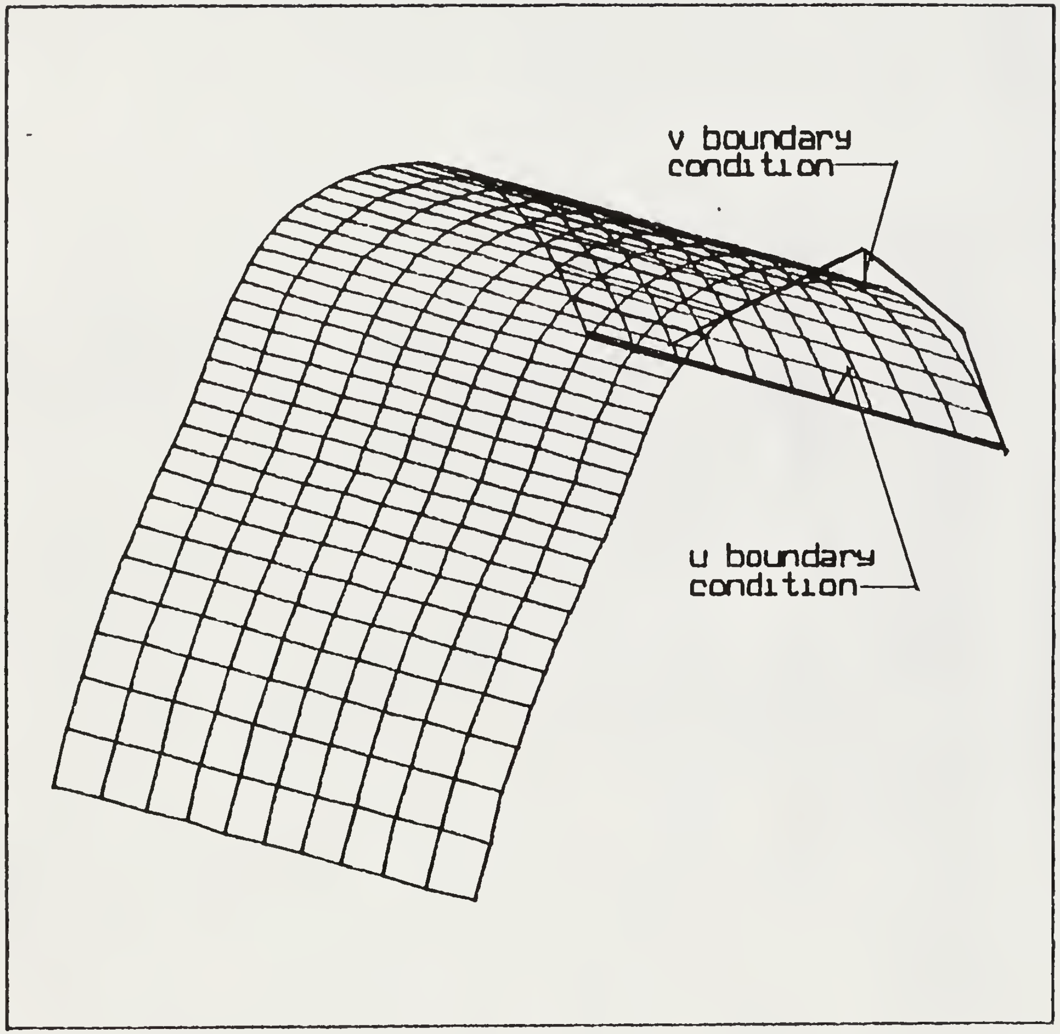


Figure 3-4   Boundary conditions for a surface.

IV. A PROGRAM TO GENERATE, MODIFY, AND DISPLAY 3-D B-SPLINE SURFACES

The Implemented Structure

The program BSPLINE is an interactive, menu guided program that can generate, display, and modify integral and rational non-periodic B-spline surfaces. There are three sets of arrays used by the program, database arrays, the display arrays, and the display file arrays. The database arrays contain the data from the data file which includes: 1) the order of the basis functions, 2) the number of the control points in each direction, 3) the knot vectors, 4) the dimension of control points and 5) the coordinate vectors of the control points. The display arrays contain information used to display the streamlines and normals to the surface in the GENERATE BSPLINE module of the program described later. The display file arrays contain: 1) the index counters needed to display a display file, 2) the geometric location of the control points, and 3) the information needed to regenerate the streamlines and normals to the surface. These are used by the module DISPLAY FILE MANAGER (to be described later) to save and read data from external storage.

The overall structure of the program is shown in figure 4-1. The main program controls access to the different modules in the program. The module GENERATE BSPLINE is used to generate the streamlines and normals and to orient the control mesh. The submodule MODIFY BSPLINE is used to locally modify B-spline control mesh. The DISPLAY FILE MANAGER is used to save and display files. Appendix B contains an index of the subroutines used by the program and a subroutine call tree organized around the program structure.

Main Program Module

BSPLINE MAIN serves as a control structure to enter the modules that have been implemented and to enter and exit the main program. The menu and presentation window are shown in Figure 4-2. The commands that are available from the main program are

GENERATE BSPLINE

DISPLAY FILE MANAGER

PRINT FILE

HELP

GENERATE BSPLINE is a module that is used to orient the control mesh for viewing and to generate the the display arrays. A more complete description will be given in a following section.

DISPLAY FILE MANAGER is used to create, save, and display the display files. A more complete description of this will also be given later.

PRINT FILE is an interactive segment that permits the user to look at the current program, database, and display arrays. This command is available from anywhere in the program and is on all menus. The primary function is to provide dynamic feedback from the program during the development of the different modules. The information displayed can also be written to system files. A different file can be created each time the command is executed.

HELP provides information for any command in the current menu by typing <HP> <menu item>. If no menu item is listed, the structure of the help command is shown. This command was not implemented in the present version of the program.

These are the commands available from the main menu of the program.

Module to Generate B-spline Surfaces

GENERATE BSPLINE is an independent module that inputs data files, orients the control mesh for viewing, and generates the arrays used to display the surface. The menu and presentation window are shown in figure 4-3. The commands available from this menu are

READ FILE

DRAW MESH

DRAW NORMALS

DRAW STREAMLINES

ISOMETRIC

ROTATE

SCALE DATA

STREAMLINES AND NORMALS

MODIFY BSPLINE

PRINT FILE

ERASE

RESET

SAVE FILE

QUIT

READ FILE and SAVE FILE are used to read and save the database data from external storage. The user specifies the <filename> and <filetype> of the external file. Examples of data files are given in Appendix C and include the orders, the knot vectors, the number of vertices in each direction, the dimension, and the coordinates of the control mesh.

STREAMLINES AND NORMALS is used to generate the streamlines and normals in the display arrays used to display the surface. The user is queried for the number of streamlines and points to generate per interval on the surface. The normals are generated at intersections of streamlines. Current values for the streamlines and normals must exist in order to use the drawing commands and to enter the MODIFY BSPLINE module of the program.

DRAW MESH, DRAW NORMALS, and DRAW STREAMLINES are used to draw the respective parts of the database and the display arrays if they exist. DRAW MESH draws the control mesh shown in figure 4-4. DRAW NORMALS draws the unit normals at the intersections of the streamlines. Figure 4-5 shows the normals. DRAW STREAMLINES draws the streamlines specified in STREAMLINES AND NORMALS. Figure 4-6 shows an example of the streamlines. Any attempt to draw something that does not exist generates a warning from the program.

ISOMETRIC, ROTATE, and SCALE DATA are used to modify the control mesh in the database for viewing. ISOMETRIC generates an isometric view by rotating the control mesh by 45 degrees about the y axis followed by 35.246 degrees about the x axis. An example of an isometric view is shown in figure 4-7. ROTATE rotates the control mesh in the database about an <axis> by an angle specified by the user. Multiple angles can be specified. The rotations are performed in the sequence given. Figure 4-8 shows a data set that has been rotated about x by 60 degrees, about y by 45 degrees and about z by 10 degrees. SCALE DATA is used to scale the control mesh in the database by a factor <scl> specified by the user. The factor must be greater than .001. Figure 4-9 shows that both the old data and the scaled data are displayed. Each time these commands are used, the user is queried whether or not to replace the current data with the modified data. If the modified data file is used the streamlines and normals must be regenerated.

MODIFY BSPLINE is a module used to locally modify the control mesh and can only be entered if current streamlines and normals exist. This module will be described in the next section.

PRINT FILE was described in the discussion of the main program.

ERASE, RESET, and QUIT are local utilities used to erase the screen, reset the database, and quit this module. Only one data file can exist in the database; therefore, the database must be reset before a new data file can be read.

Display File Module

DISPLAY FILE MANAGER is an independent module that is used for the display of the control mesh, streamlines and normals. It both reads and saves display files from memory or creates them from the display arrays used in GENERATE BSPLINE if they exist. The menu and presentation window are shown in figure 4-10. The commands available from this menu are:

CREATE FILE
READ FILE
DRAW MESH
DRAW STREAMLINES
DRAW NORMALS
SCREEN TYPE
SCALE OUTPUT
PRINT FILE
ERASE
RESET

SAVE FILE

QUIT

CREATE FILE creates a display file by copying the control mesh, the display arrays, and the counters needed to display them into the display file arrays from GENERATE BSPLINE arrays. These arrays can then be saved on external media.

READ FILE and SAVE FILE are used to read and save display files to and from external storage. The data file can not be constructed from the data in the display file.

DRAW MESH, DRAW NORMALS, DRAW STREAMLINES are similar to the commands given in GENERATE BSPLINE except they use the display file arrays.

SCREEN TYPE permits the user to specify the presentation window as either the standard window with menu or a full screen without menu. Figure 4-11 shows a full screen display.

SCALE OUTPUT scales the size of the output window when using the full screen mode. Figure 4-12 shows a 20% reduction in the size of the window.

PRINT FILE is the same as in the main program.

ERASE, RESET, AND QUIT are utilities used to clear the screen, initialize the display arrays, and quit this module.

These are the commands available in the display module.

General Information

The program consists of approximately 7,000 lines of commented FORTRAN 77 source code. It was written on the Kansas State University Computing Center's NAS 6630 mainframe using a subset of PLOT 10 IGL graphics subroutines. The output devices were a Selenar Hirez 100 graphics terminal setup to dump to a Hewlett/Packard 7470A plotter.

In this chapter the operation of the MAIN program, GENERATE BSPLINE, and DISPLAY FILE MANAGER modules have been explained and illustrated. The next chapter explains the operation of the MODIFY BSPLINE module.

```
BSPLINE MAIN
MODULE
GENERATE BSPLINE <GNB>+++++++
DISPLAY FILE MANAGER <DFM>++|+++++++++++++++++++++++++++++
PRINT FILE <PF>             |                            |
HELP <HP>                   |                            |
QUIT PROGRAM <QT>           |                            |
                            |                            |
                            V                            V
                            GENERATE BSPLINE             DISPLAY FILE
                            MODULE                       MANAGER MODULE
                            READ FILE <RF>               CREATE FILE <CR>
                            DRAW MESH <DM>               READ FILE <RF>
                            DRAW NORMALS <DN>            DRAW MESH <DM>
                            DRAW STREAMLINES <DS>        DRAW
                            ISOMETRIC <IS>               STREAMLINE <DS>
                            ROTATE <RT>                  SCREEN TYPE <ST>
                            SCALE DATA <SD>              PRINT FILE <PF>
                            STREAMLINES AND              ERASE <ER>
                            NORMALS <SN>                 RESET <RS>
                            MODIFY BSPLINE <MDB>++++     SAVE FILE <SV>
                            PRINT FILE <PF>        |     QUIT <QT>
                            ERASE <ER>             V
                            RESET <RS>             MODIFY BSPLINE
                            SAVE FILE <SF>         MODULE
                            QUIT <QT>              MOVING POINTER <MP>
                                                   DRAW ACTIVE MESH <DA>
                                                   DRAW MESH <DM>
                                                   DRAW STREAMLINES <DS>
                                                   MOVE PIJ"S <MV>
                                                   UPDATE <UP>
                                                   PRINT FILE <PF>
                                                   ERASE <ER>
                                                   QUIT <QT>
```

Figure 4-1 Program structure

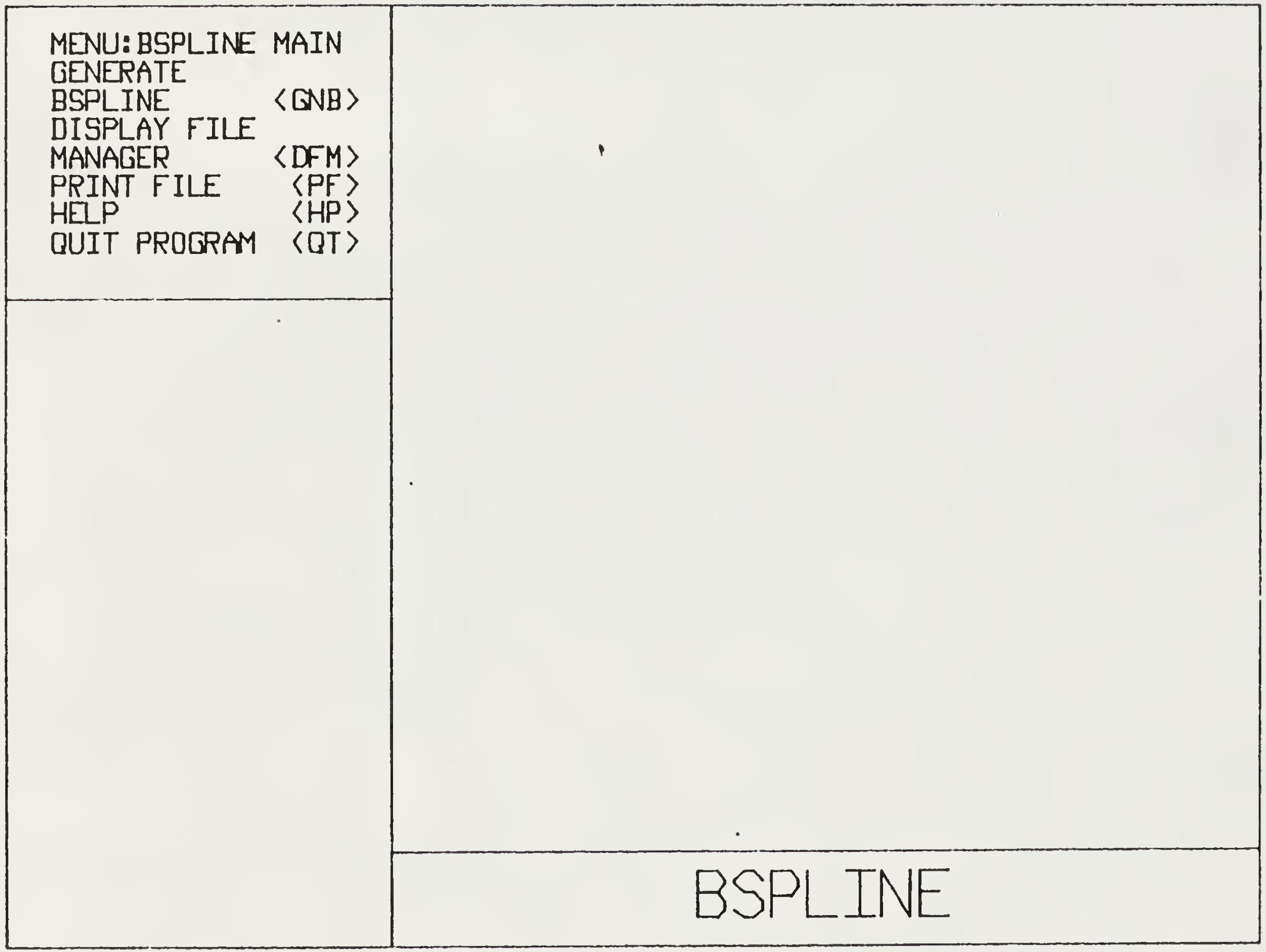


Figure 4-2 Menu and presentation window for BSPLINE MAIN.

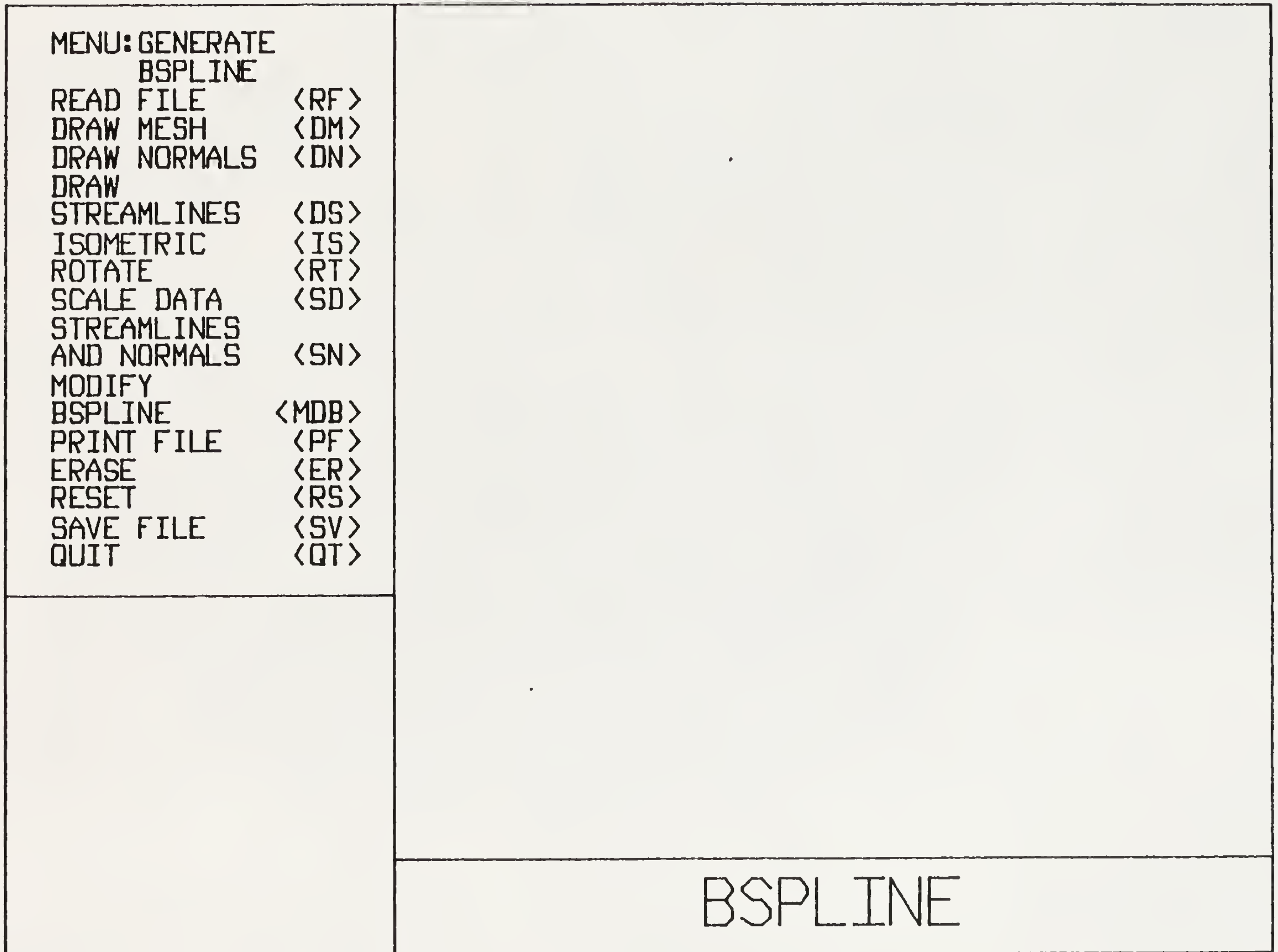


Figure 4-3 Menu and presentation window for GENERATE BSPLINE.

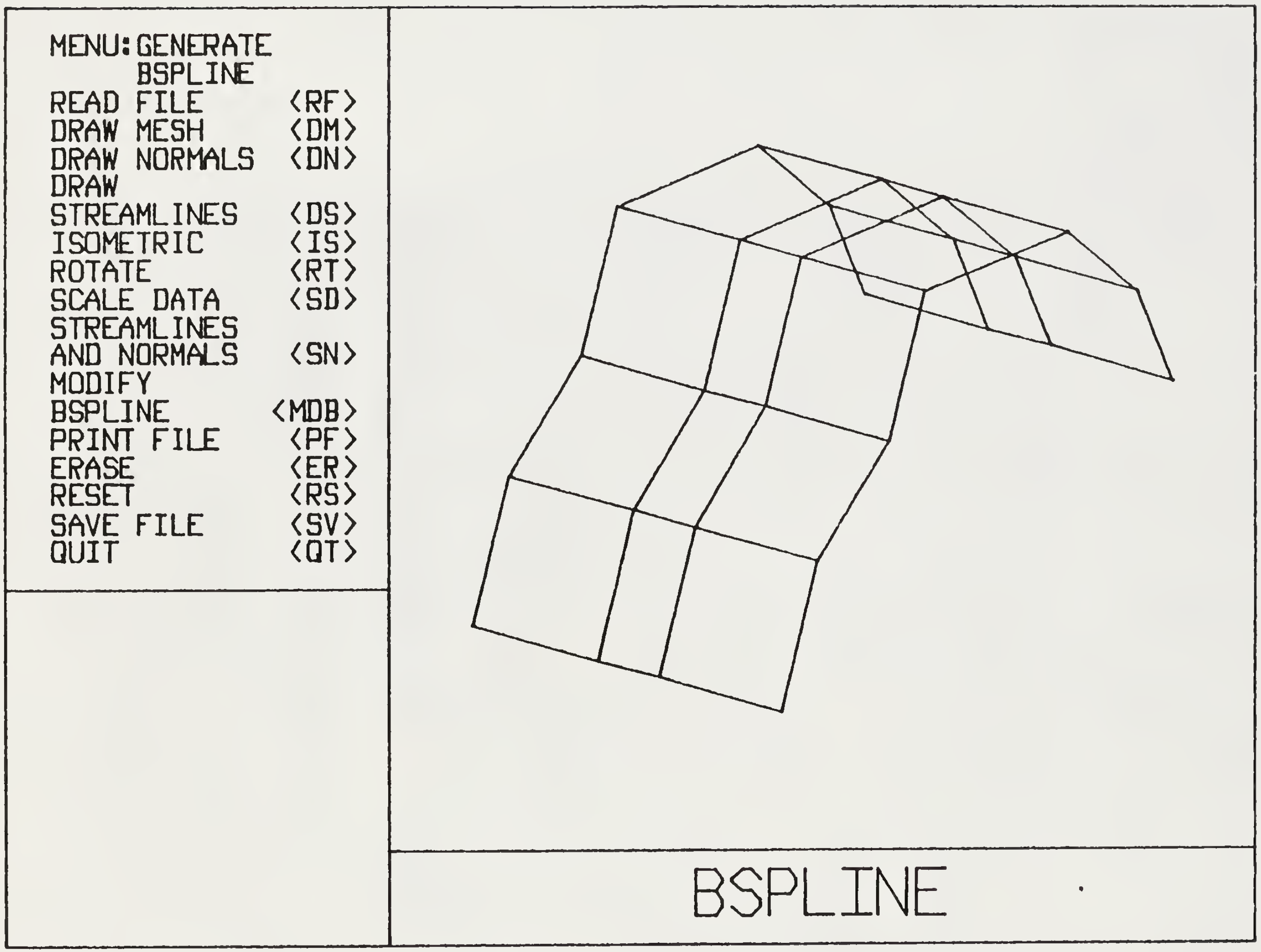


Figure 4-4 DRAW MESH in module GENERATE BSPLINE.

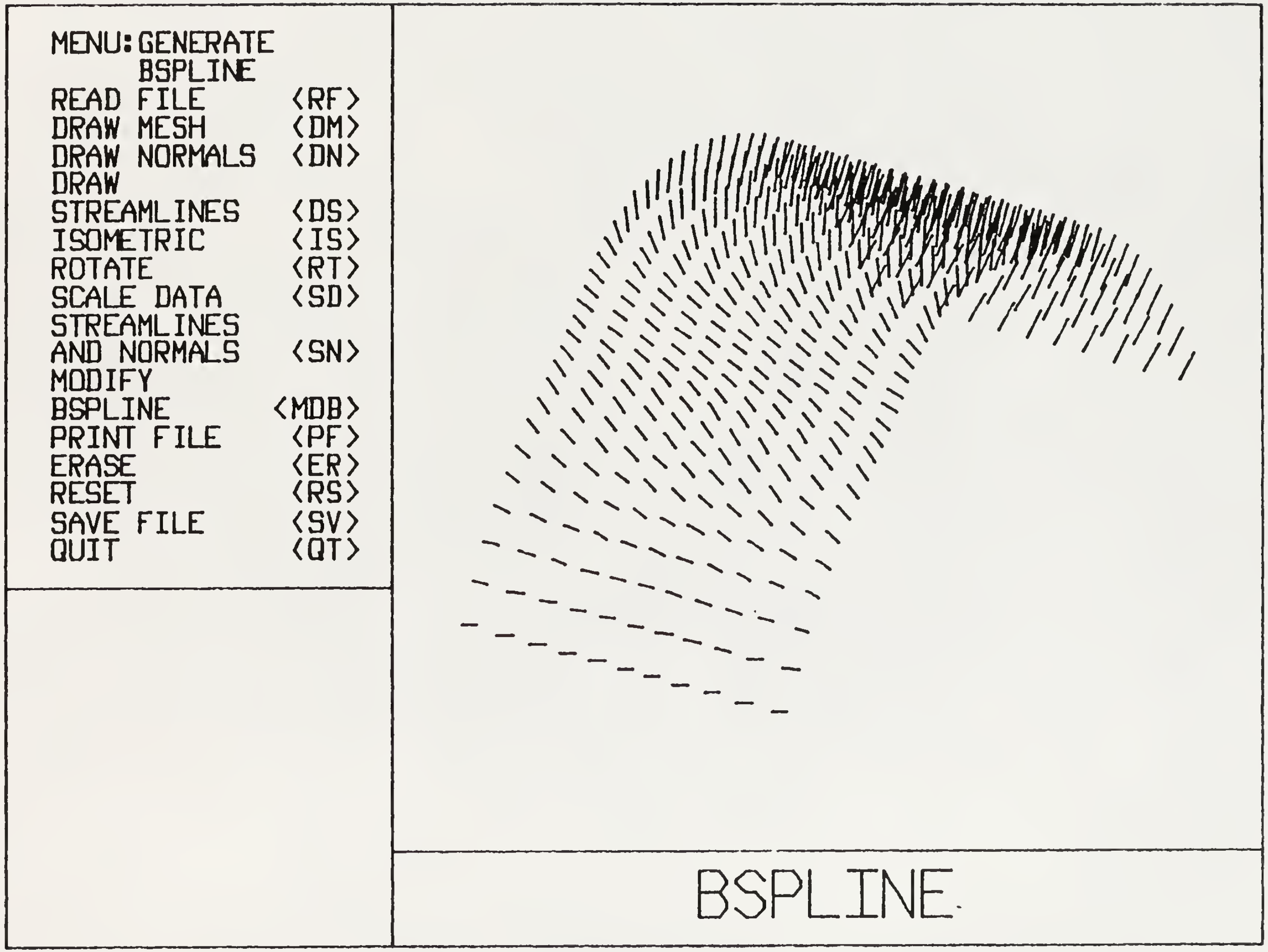


Figure 4-5 DRAW NORMALS in module GENERATE BSPLINE.

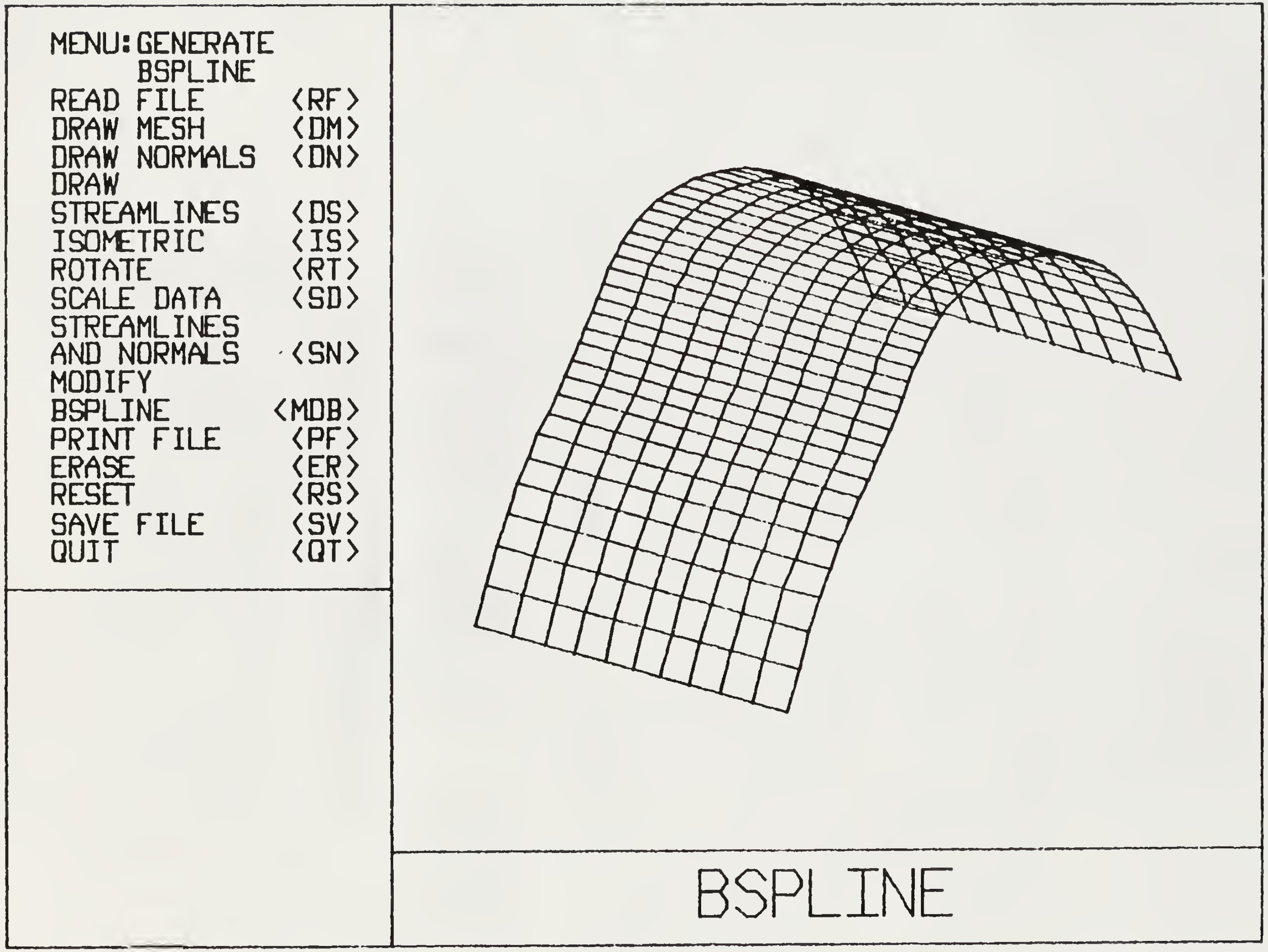


Figure 4-6 DRAW STREAMLINES in module GENERATE BSPLINE.

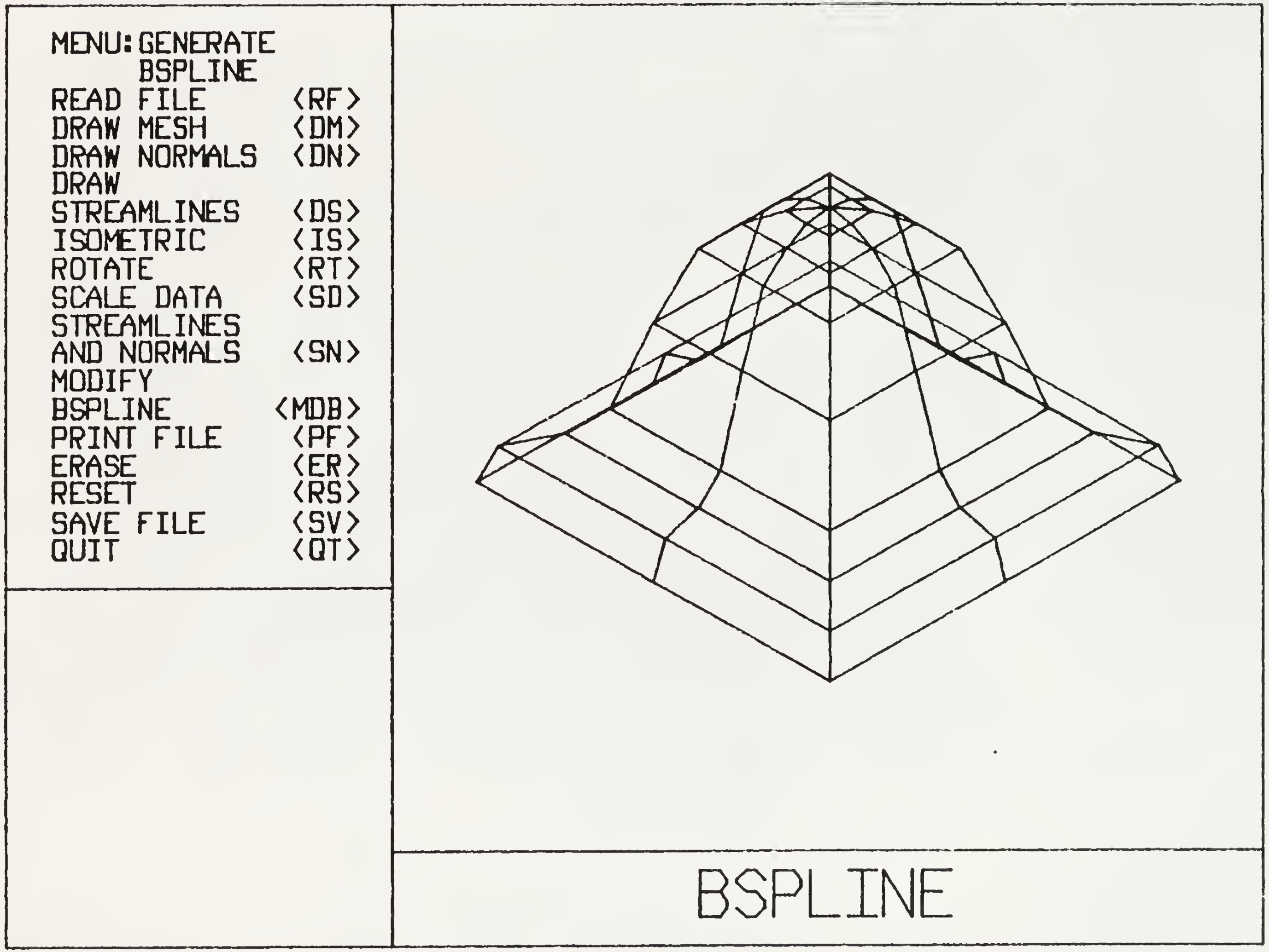


Figure 4-7 ISOMETRIC in module GENERATE BSPLINE.

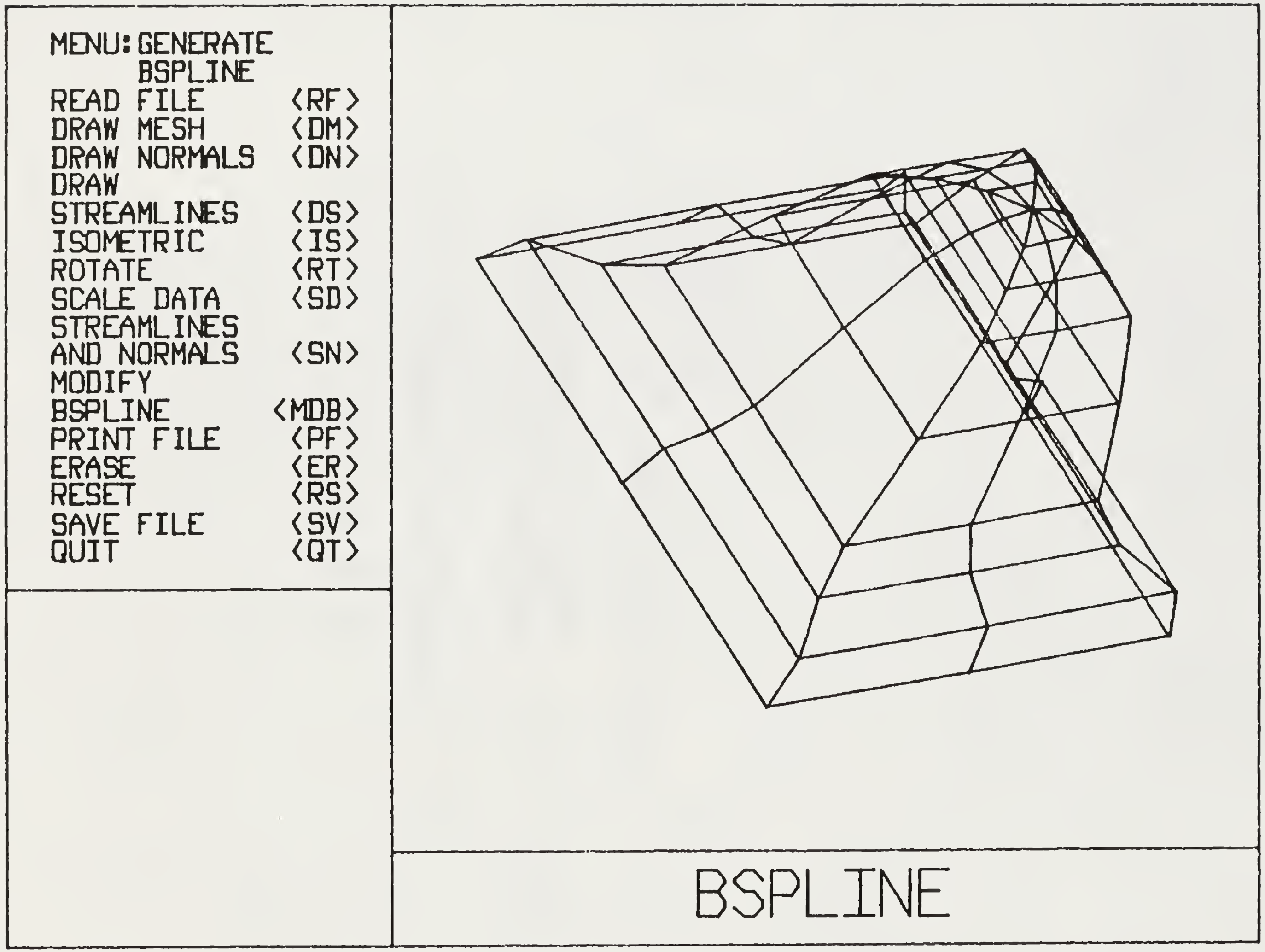


Figure 4-8 ROTATE in module GENERATE BSPLINE.

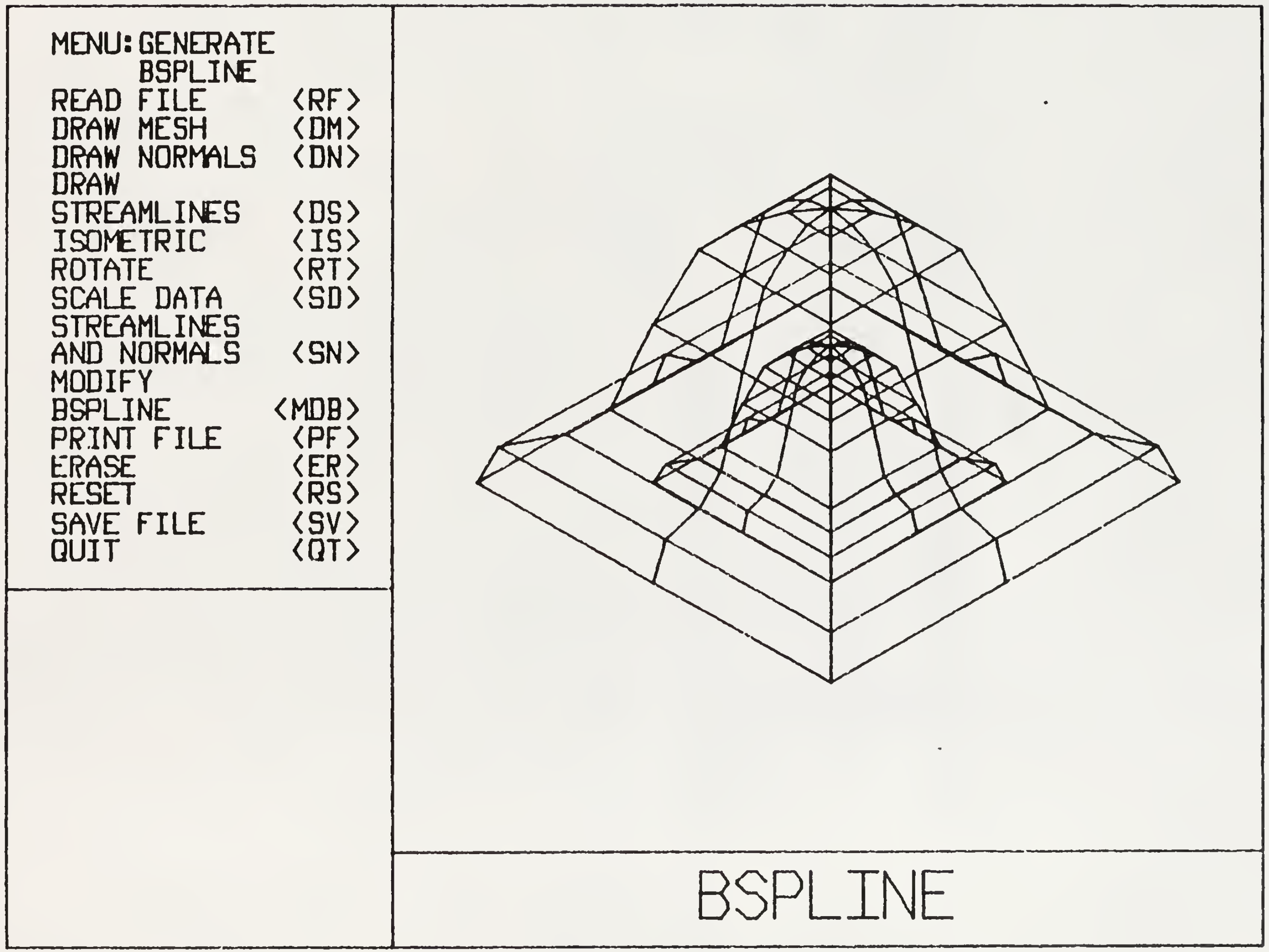


Figure 4-9 SCALE DATA in module GENERATE BSPLINE.

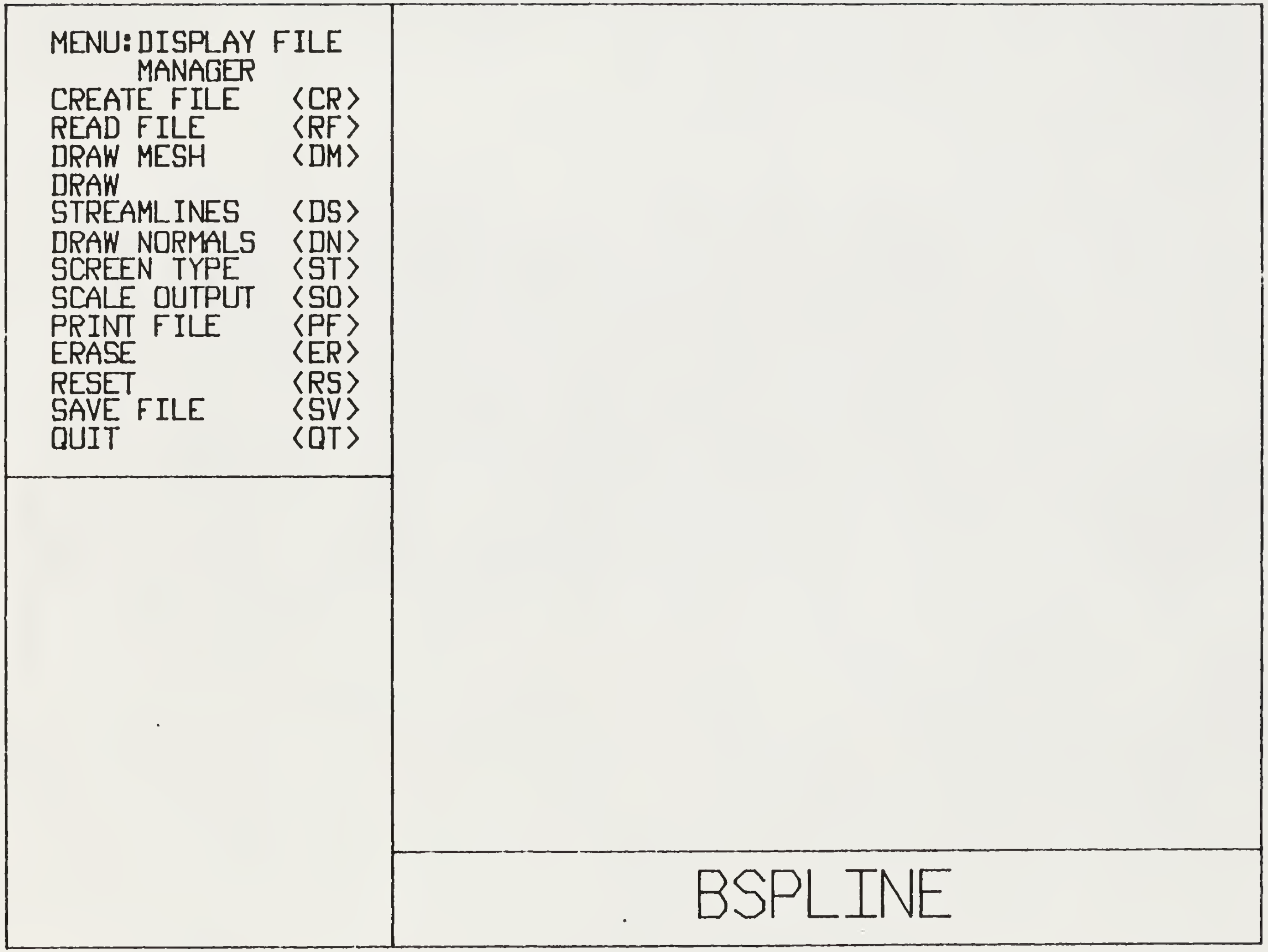


Figure 4-10 Menu and presentation window for DISPLAY FILE MANAGER.

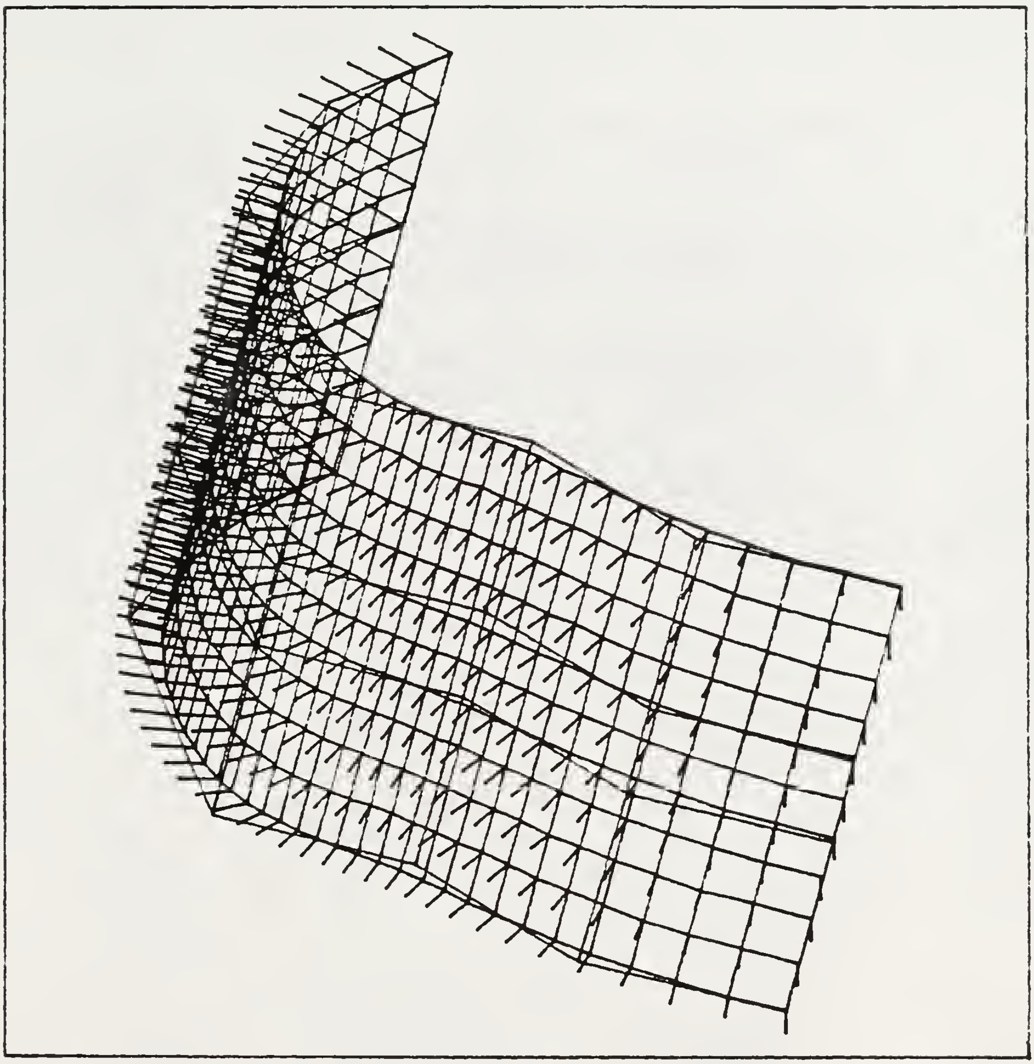


Figure 4-11 SCREEN TYPE in module DISPLAY FILE MANAGER.

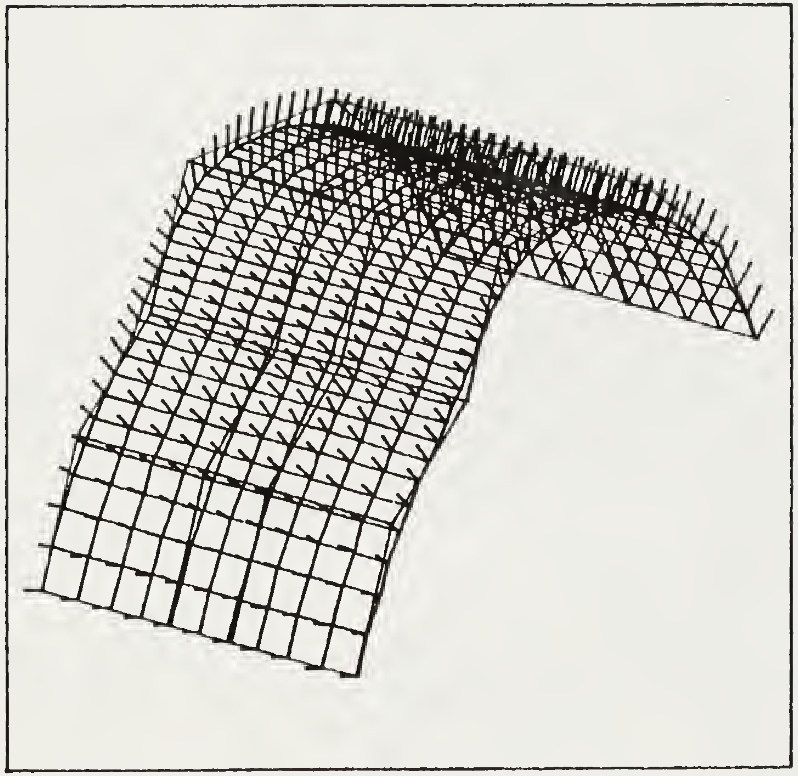


Figure 4-12 SCALE OUTPUT in module DISPLAY FILE MANAGER.

V. INTERACTIVE MODIFICATION OF B-SPLINE SURFACES

MODIFY BSPLINE is a module that can be accessed from GENERATE BSPLINE when current streamlines and normals exist. The module is used to locally modify the surface by moving the points of the control mesh in space. The menu and presentation window are shown in figure 5-1. The commands available from this menu are:

MOVING POINTER

DRAW ACTIVE MESH

DRAW MESH

DRAW STREAMLINES

MOVE PIJ"S

UPDATE

PRINT FILE

ERASE

QUIT

Relevant Geometric Properties

The geometric properties that are used in this module are the local support, the convex hull, and the existence of a u,v surface coordinate system.

Local support guarantees that there will be a finite mesh of control points associated with the generation of any point on the surface. The active control mesh will be an array of control points no larger than the order of the surface in the u direction by the order of the surface in the v direction.

The convex hull property of the B-spline surface guarantees that any point on the surface will be in the space of the convex hull of the polyhedron, defined by the active control mesh.

A u,v coordinate system results from the use of the parametric representation of the surface. The parameters map the three-dimensional x, y, z geometric space onto the two-dimensions of the parameters u and v. This mapping permits the effective use of two dimensional input devices such as thumb wheels, cursor keys, or a mouse to move a place marker on the surface.

The combination of these properties led to the development of an effective means of positioning a cursor on the surface. Associated with that position is an active control mesh, with which to modify a that portion of the curve.

Critical Commands

MOVING POINTER is used to move a surface pointer on the u,v surface and to dynamically display the active control mesh associated with the point. Pointer position is controlled using four cursor keys, +u, -u, +v, -v.

The one-to-one relationship between a point on the surface and the active control mesh required to generate it permits the dynamic display of the active control mesh. Shown in Figure 5-2. Both the parametric and the Cartesian location are displayed and updated with each movement of the pointer. The active control mesh is displayed when DRAW ACTIVE MESH is on. The purpose of MOVING POINTER is to position the pointer on the surface and establish the mesh that can be modified in MOVE PIJ"S.

MOVE PIJ"S is used to change the location in space of the elements of the active control mesh. The active control mesh is determined by the location of surface pointer in MOVING POINTER. An interactive sequence precedes the modification of control points where

1. The user chooses whether to display the entire control mesh or zoom the active control mesh.

    If the user chooses to zoom the active control mesh then only the active control mesh and the adjacent control points are displayed.

    (Figures 5-4 and 5-5 illustrate this difference.)

2. The user sets the mover sensitivity, a control which adjusts the incremental amount the points are moved for each stroke of the control keys. The default is .01 of the maximum coordinate difference in the x, y, z, directions that the mesh occupies.
3. The labels are listed, and the user chooses one of the elements of the active control mesh to modify by specifying its I,J mesh index.

Six control keys are used to change the position of the selected point: +x, -x, +y, -y, +z, -z. Multiple points can be changed, or the user may exit the command.

UPDATE is used to recalculate the streamlines and normal after the control mesh is modified. The point and interval counters are the same as those used in STREAMLINES AND NORMALS in GENERATE BSPLINE module. The display arrays must be updated after any modification of the control mesh to be able to display the streamlines.

Other Commands

DRAW ACTIVE MESH is used to activate the dynamic drawing of the active mesh in MOVING POINTER.

DRAW MESH and DRAW STREAMLINES are used to draw the current mesh and streamlines.

PRINT FILE is the same as in the main program.

ERASE is a utility to erase the screen.

QUIT quits this submodule and returns to GENERATE BSPLINE.

This completes the description of the module to locally modify the B-spline control mesh. Next is a review of what has been done with possible recommendation for future work.

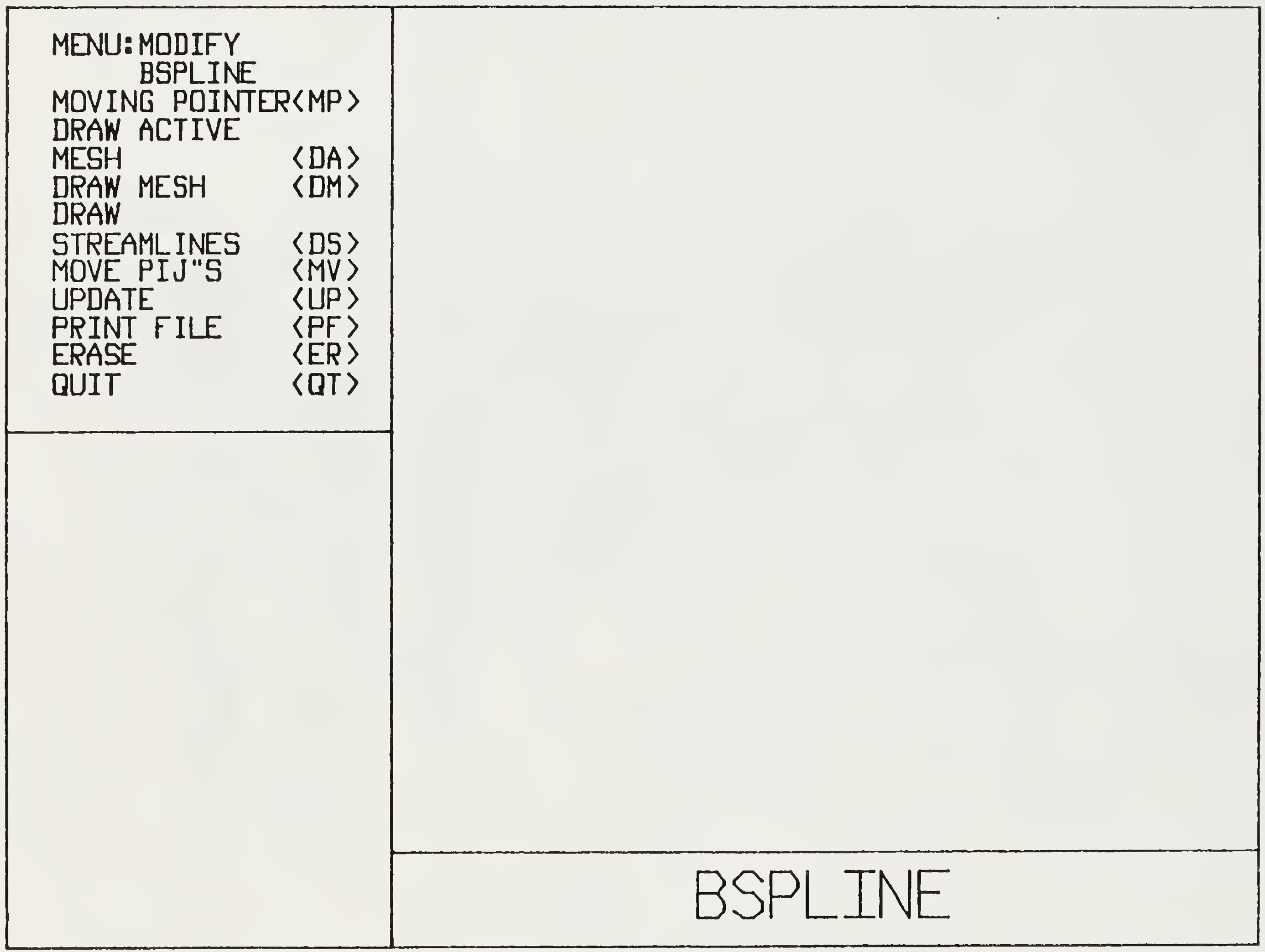


Figure 5-1 Menu and presentation window for MODIFY BSPLINE.

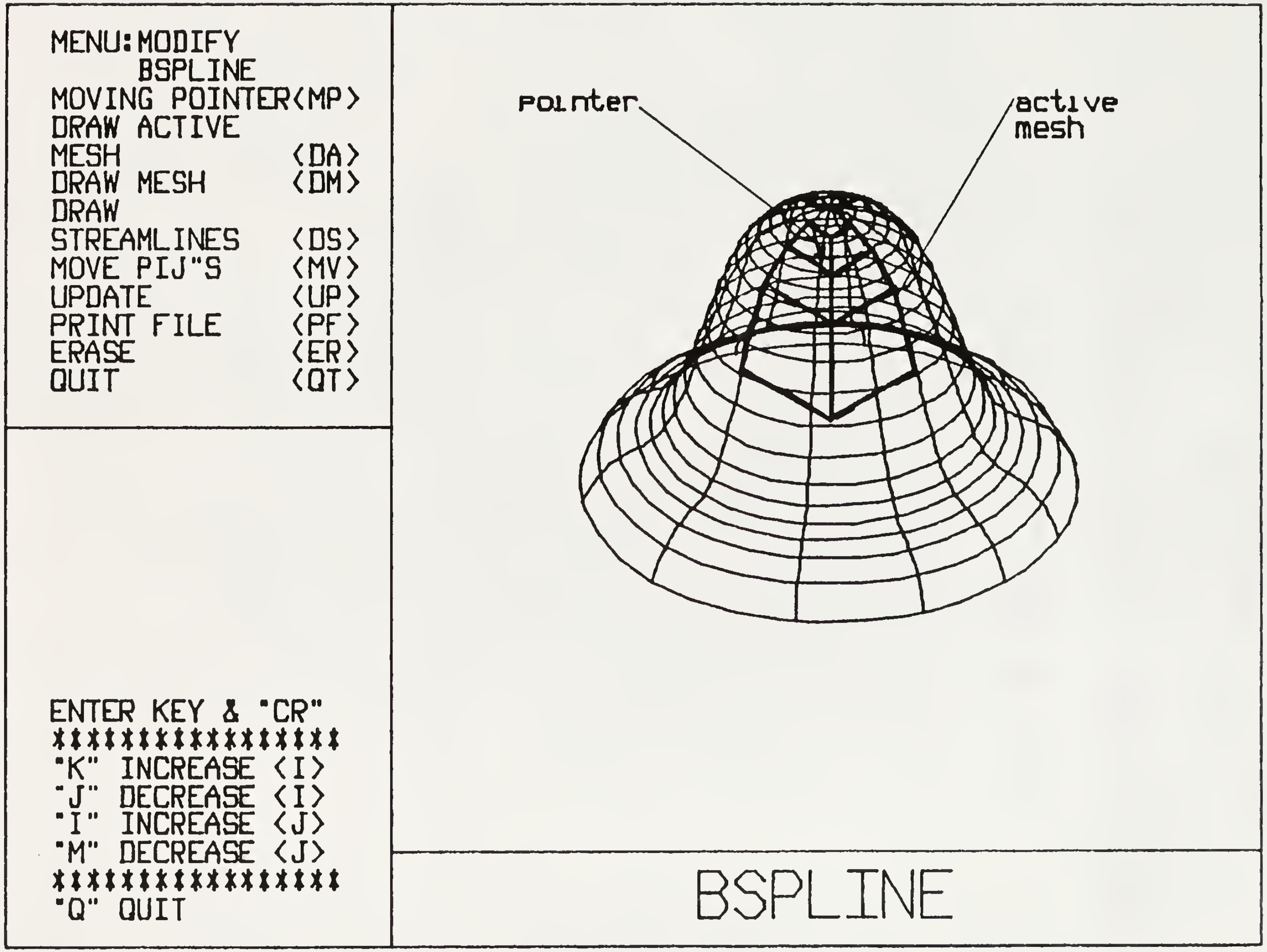


Figure 5-2 MOVING POINTER in module MODIFY BSPLINE.

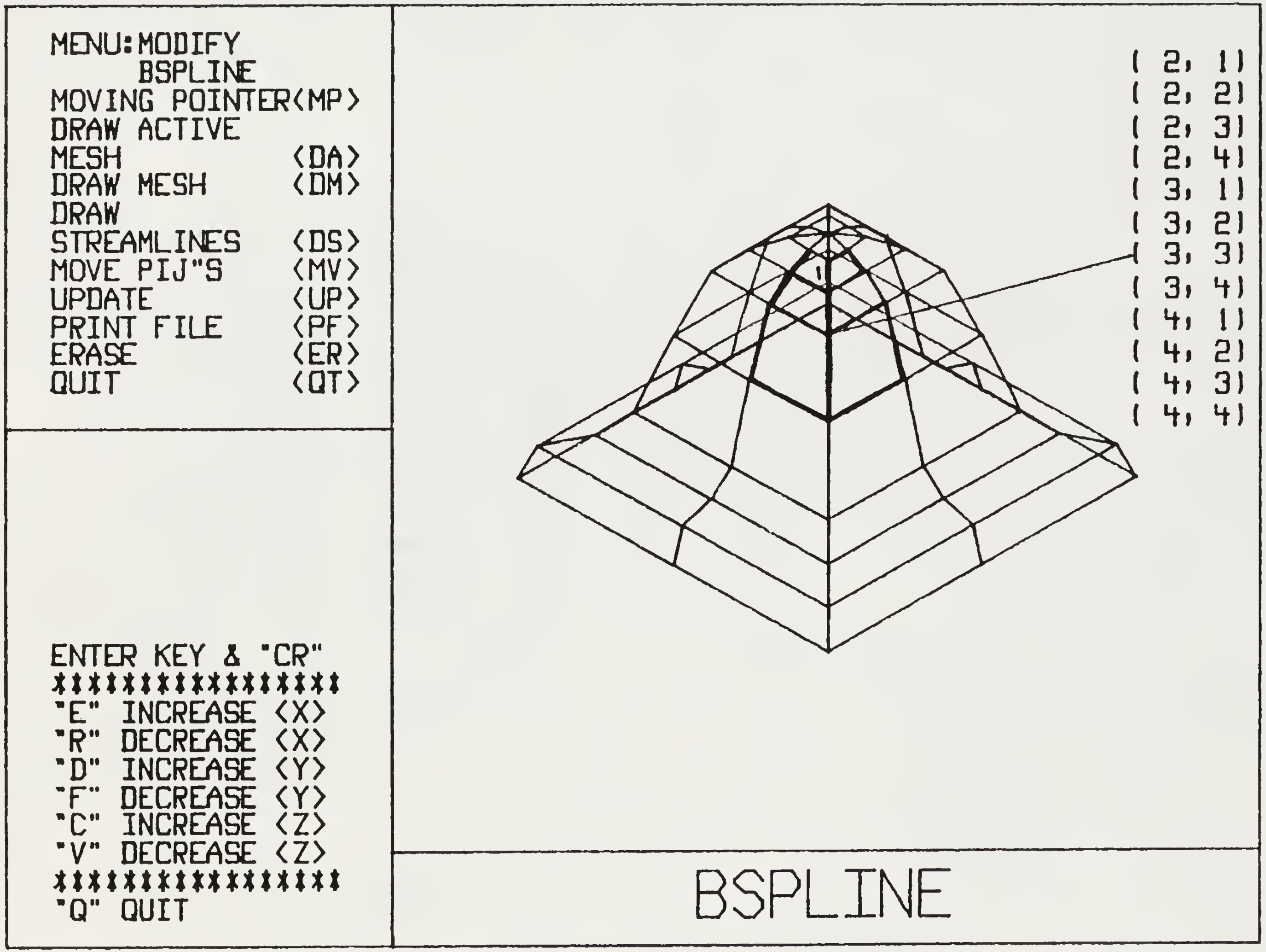


Figure 5-3 MOVE PIJ"S in module MODIFY BSPLINE.

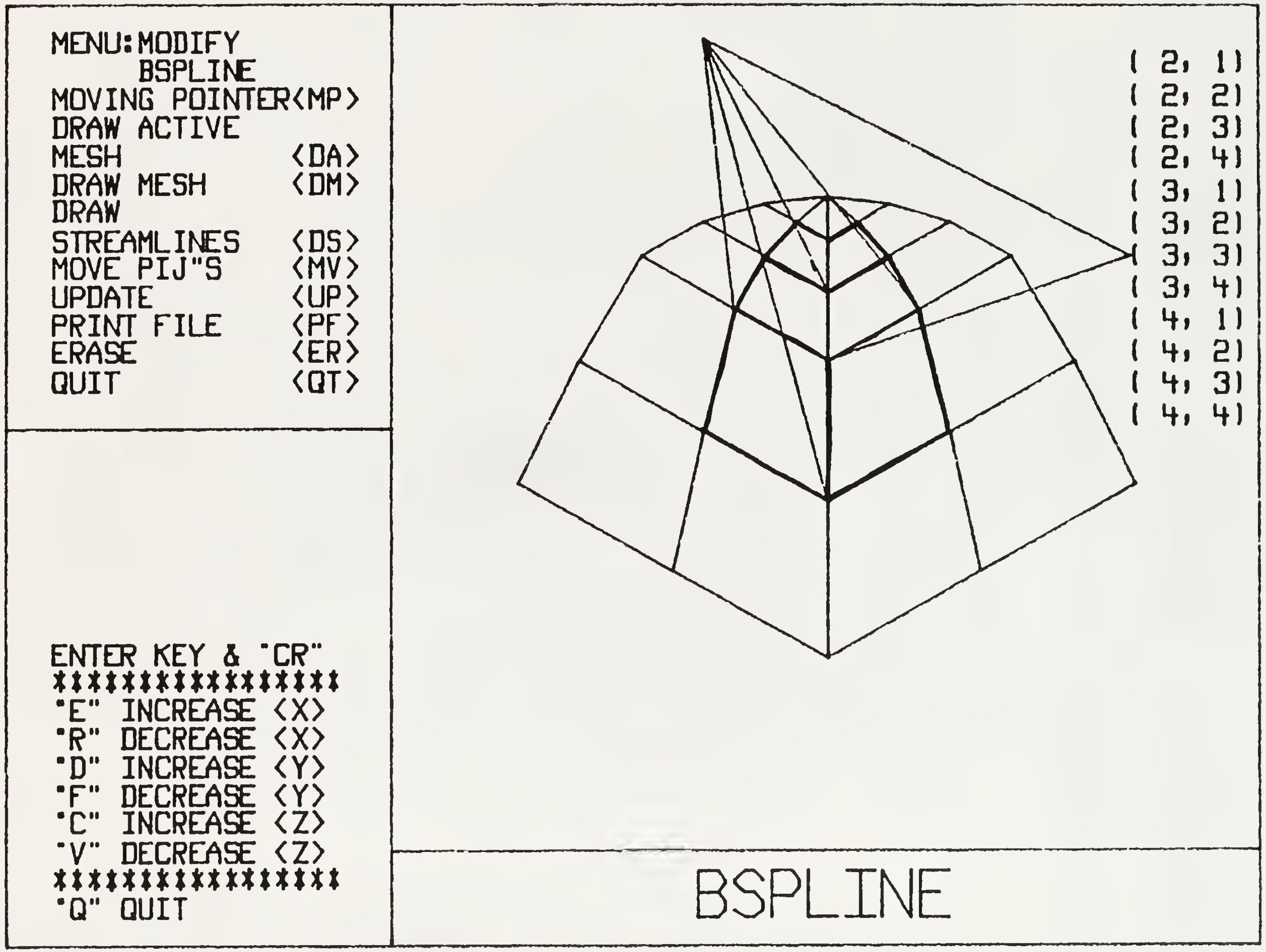


Figure 5-4 MOVE PIJ"S with zoom in module MODIFY BSPLINE.

VI. CONCLUSIONS

Accomplishments of this present work

There were three main accomplishments in this work: 1) the researching of the B-spline surface representations, 2) the development of an interactive program to generate and display B-spline surfaces, and 3) the implementation of a method for local modification of B-spline surfaces.

The research involved an extensive review of both current and historic literature about the application of the B-spline approximation to the computer generation of surface models. Using the information aquired in the literature search, an interactive program was written to generate and display both integral and rational B-spline models. Finally, to extend the usefulness of the B-spline representation, a method of locally modifying B-spline surfaces was developed. The basis for the development was the local support and convex hull properties. The method utilized the inherent two-dimensional characteristics of the parametric representation to move a location pointer on the surface. Using the location of the parametric pointer to specify a finite set of control points, an intuitive method of modifying the shape of the curve was developed.

Possible Extensions of this Work

Some directions that could be taken in extending this work are: further development of the program, use of the present program as an educational tool, and the development of applications based on the surface representation.

Other modules that would be useful are

1. an interactive database builder that would permit the construction of original B-spline surfaces
2. a module to permit the modeling of multiple surfaces at the same time
3. a module for large scale reshaping of the control mesh
4. the development of a more sophisticated display module to include hidden surface removal and shadowing algorithms to be used when displaying surfaces.

The present program and thesis could be used as an educational tool to introduce students to the characteristic of the B-spline surface and to the algorithms required to generate them. The generate module could be used in another program to model approximate multi-dimensional data sets.

Various modules developed in this work could be used individually or together in graphical applications for mechanical part design, analysis, or manufacture.

The B-spline representation for graphical entities is a powerful mathematical tool in the development of engineering analysis and design.

REFERENCES

1. C. deBoor, "On Calculating with B-Splines", J. Approximation Theory, Vol 6, pp. 50-62, 1972

2. R. F. Riesenfeld, "Application of B-Spline Approximation to Geometric Problems of Computer Aided Design", PhD Dissertation, Computer Science Department, Syracuse University, Syracuse NY, May 1973

3. I.J. Schoenberg, "Contributions to the Approximation of Equidistant Data By Analytic Functions", Q. Appl. Math., Vol 4, pp. 45-99;112-141, 1946

4. K.J. Verspille, "Computer Aided Design Application of Rational B-Spline Approximation Forms" PhD Dissertation, Computer Science Department, Syracuse University, Syracuse NY, February 1975

5. D.F. Rodgers and J.A. Adams, "Mathematical Elements for Computer Graphics", McGraw-Hill, New York, 1976

6. W. Tiller, "Rational B-Splines for Curve and Surface Representation", IEEE Computer Graphics and Applications, pp. 61-69, September 1983

7. G.B. Thomas Jr., "Calculus and Analytic Geometry", 4th edition, Addison-Wesley, December 1969

BIBLIOGRAPHY

1. J. Alander, M. Mantyla, T. Ratanen, "Solid Modeling with Parametric Surfaces", Eurographics 83, edited by P.J.w. Ten Hagen, North Holland Publishing Co., Amsterdam 1983

2. J.C. Beatty, R.H. Bartels, B.A. Barsky "An Introduction to the Use of Splines in Computer Graphics", University of Waterloo TR-CS-83-09 Waterloo, Ontario, Canada 1983

3. B.A. Barsky, "End Conditions and Boundary Conditions for Uniform B-Spline Curve and Surface Representations", Computers in Industry, Vol 3, No 3, pp. 17-29, March 1982

4. R.H. Bartels, M.A.J. Sweeney, "Ray Tracing Free Form B-Spline Surfaces", IEEE Computer Graphics and Applications, Vol 6, No 2, pp. 41-49, February 1986

5. R.M. Blomgren, "Rational B-Spline Curves and Surfaces", Boeing Document D-441.1

6. W. Boehm, "Inserting New Knots into B-spline Curves", Computer Aided Design, Vol 12, No 4, pp. 199-201, July 1980

7. E. Catmull, J. Clark, "Recursively Generated B-Spline Surfaces on Arbitrary Topological Meshes", Computer Aided Design, Vol 10, No 6, pp. 350-355, November 1978

8. E. Cohen, T. Lyche, L.L. Schumaker, "Algorithms for Degree Raising in B-Spline Representation for a General Curve", Computers and Graphics Vol 7, No 4, pp. 172-181, July 1985

9. S.A. Coons, "Surface Patches and B-Spline Curves", Computer Aided Geometric Design edited by R. Barnhill and R.F. Riesenfeld, Academic Press 1974

10. C. deBoor, "A Practical Guide to Splines", Springer and Verlag, New York, 1978

11. A.R. Forrest, "On Coons and Other Methods for Representation of Curved Surfaces", Tutorial and Selected Reading in Interactive Computer Graphics, IEEE Computer Society, 1979, reprinted from Computer Graphics and Image Processing, 1972

12. C. Frost, G. Kinzel, "An Automatic Adjustment Procedure for Rational Splines", Computers and Graphics, Vol 6, No 4, pp 171-176, 1982

13. W.K. Giloi, "Interactive Computer Graphics", Prentice-Hall, 1978

14. R.N. Goldman, T.W. Sederberg, "Algebraic Geometry for Computer Aided Design", IEEE Computer Graphics and Applications, Vol 6, no 6, pp. 52-59, June 1986

15. W.J. Gordon, R.F. Riesenfeld, "B-Spline Curves and Surfaces", from Computer Aided Geometric Design edited by R. Barnhill and R.F. Riesenfeld, Academic Press, 1974

16. T.N.E. Greville, "Theory and Application of Spline Functions", Proceedings of an Advanced Seminar for the US Army October 7-9, 1968, Mathematics Research Center, University of Wisconsin, edited by T.N.E. Greville, Academic Press, New York, 1969

17. P.J. Hartley, C.J. Judd, "Parameterization and Shape of B-Spline Curves for CAD", Computer Aided Design, Vol 12, No 5, pp. 235-238, September 1980

18. O. Lozover, K. Preiss, "Automatic Construction of a Cubic B-Spline Representation for a General Curve", Computers and Graphics, Vol 7, No 2, pp. 149-153, 1983

19. W.M. Newman, R.F. Sproull, "Principles of Interactive Computer Graphics", 2nd edition, McGraw-Hill, New York, 1979

APPENDIX A
B-spline Basis Function Evaluations

B-SPLINE FUNDEMENTAL EVALUATIONS

$F(t)=sumover(i:j-(K-1),j)\ A_i * N_{i,K}(t) \quad : j \text{ s.t. } t_j <= t < t_{j+1}$

$N_{i,1}(t)= \begin{cases} 1.0 & \text{if } t_i <= t < t_{i+1} \\ & \text{else } 0.0 \end{cases}$

$$N_{i,K}(t)= \frac{t - t_i}{t_{i+K-1} - t_i} N_{i,K-1}(t) + \frac{t_{i+K} - t}{t_{i+K} - t_{i+1}} N_{i+1,K-1}(t)$$

If K=1 then sumover(i:j-(K-1),j)=sumover(i:j,j)

j s.t. $t_j <= t < t_{j+1}$

therefore $N_{i,1}(t)$; not equal to 0.0

****

$N_{i,1}(t)= 1.$

****

If K=2 then sumover(i:j-(K-1),j)=sumover(i:j-1,j)

j s.t. $t_j <= t < t_{j+1}$

therefore $N_{i-1,2}(t);N_{i,2}(t)$; not equal to 0.0

Note: i=i-1 in the general definition.

$$N_{i-1,2}(t) = \frac{t - t_{(i-1)}}{t_{(i-1)+1} - t_{(i-1)}} N_{(i-1),1} + \frac{t_{(i-1)+2} - t}{t_{(i-1)+2} - t_{(i-1)+1}} N_{(i-1)+1,1}(t)$$

|
V

$$= \frac{t - t_{(i-1)}}{t_i - t_{(i-1)}} N_{(i-1),1}(t) + \frac{t_{i+1} - t}{t_{i+1} - t_i} N_{i,1}(t)$$

****

$$\mathbf{N_{i-1,2}(t)} = \frac{t_{i+1} - t}{t_{i+1} - t_i} N_{i,1}(t)$$

****

Note: i=i in the general definition.

$$N_{i,2}(t) = \frac{t - t_i}{t_{i+1} - t_i} N_{i,1}(t) + \frac{t_{i+2} - t}{t_{i+2} - t_{i+1}} N_{i+1,1}(t)$$

****

$$\mathbf{N_{i,2}(t)} = \frac{t - t_i}{t_{i+1} - t_i} N_{i,1}(t)$$

****

If K=3 then sumover(i:j-(K-1),j)=sumover(i:j-2,j)

j s.t. t_j <= t < t_{j+1}

therefore $N_{i-2,3}(t)$; $N_{i-1,3}(t)$; $N_{i,3}(t)$; are not equal to 0.0 .

Note: i=i-2 in the general definition.

$$N_{i-2,3}(t) = \frac{t - t_{(i-2)}}{t_{(i-2)+2} - t_{(i-2)}} N_{(i-2),2}(t) + \frac{t_{(i-2)+3} - t}{t_{(i-2)+3} - t_{(i-2)+1}} N_{(i-2)+1,2}(t)$$

|
V

$$= \overset{\textbf{I(i-2)}}{\frac{t - t_{(i-2)}}{t_i - t_{(i-2)}}} \overset{\textbf{N(i-2),2}}{N_{(i-2),2}(t)} + \overset{\textbf{II(i-2)}}{\frac{t_{i+1} - t}{t_{i+1} - t_{i-1}}} \overset{\textbf{N(i-1),2}}{N_{i-1,2}(t)}$$

$$N_{i-2,3}(t) = \overset{\textbf{I(i-2)}}{\frac{t - t_{(i-2)}}{t_i - t_{(i-2)}}} \overset{\textbf{N(i-2),2}}{\left[\frac{t - t_{(i-2)}}{t_{(i-2)+1} - t_{(i-2)}} N_{(i-2),1}(t) + \frac{t_{(i-2)+2} - t}{t_{(i-2)+2} - t_{(i-2)+1}} N_{(i-2)+1,1}(t) \right]}$$

|
V

$$= \overset{\textbf{I(i-2)}}{\frac{t - t_{(i-2)}}{t_i - t_{(i-2)}}} \overset{\textbf{N(i-2),2}}{\left[\frac{t - t_{(i-2)}}{t_{i-1} - t_{(i-2)}} N_{(i-2),1}(t) + \frac{t_i - t}{t_i - t_{i-1}} N_{i-1,1}(t) \right]}$$

$$+ \overset{\textbf{II(i-2)}}{\frac{t_{(i-2)} - t}{t_{i+1} - t_{i-1}}} \overset{\textbf{N(i-1),2}}{\left[\frac{t_{i+1} - t}{t_{i+1} - t_i} N_{i,1}(t) \right]}$$

****

$$N_{i-2,3}(t) = \overset{\textbf{II(i-2)}}{\frac{t_{i+1} - t}{t_{i+1} - t_{i-1}}} \overset{\textbf{N(i-1),2}}{\left[\frac{t_{i+1} - t}{t_{i+1} - t_i} N_{i,1}(t) \right]}$$

****

Note: i=i-1 in the general definition

$$N_{i-1,3}(t) = \frac{t - t_{(i-1)}}{t_{(i-1)+2} - t_{(i-1)}} N_{(i-1),2}(t) + \frac{t_{(i-1)+3} - t}{t_{(i-1)+3} - t_{(i-1)+1}} N_{(i-1)+1,2}(t)$$

$$= \underbrace{\frac{t - t_{(i-1)}}{t_{i+1} - t_{(i-1)}}}_{\text{I(i-1)}} \underbrace{N_{(i-1),2}(t)}_{\text{N(i-2),2}} + \underbrace{\frac{t_{i+2} - t}{t_{i+2} - t_i}}_{\text{II(i-1)}} \underbrace{N_{i,2}(t)}_{\text{N i,2}}$$

****

$$N_{i-1,3}(t) = \underbrace{\frac{t - t_{(i-1)}}{t_{i+1} - t_{(i-1)}}}_{\text{I(i-1)}} \underbrace{\left[\frac{t_{i+1} - t}{t_{i+1} - t_i} N_{i,1}(t) \right]}_{\text{N(i-1),2}} + \underbrace{\frac{t_{i+2} - t}{t_{i+2} - t_i}}_{\text{II(i-1)}} \underbrace{\left[\frac{t - t_i}{t_{i+1} - t_i} N_{i,1}(t) \right]}_{\text{N i,2}}$$

****

Note: i=i in the general definition

$$N_{i,3}(t) = \overset{I(i)}{\frac{t - t_i}{t_{i+2} - t_i}} \; \overset{N\,i,2}{N_{i,2}(t)} + \overset{II(i)}{\frac{t_{i+3} - t}{t_{i+3} - t_{i+1}}} \; \overset{N(i+1),2}{N_{i+1,2}}$$

$$= \overset{I(i)}{\frac{t - t_i}{t_{i+2} - t_i}} \left[\overset{N\,i,2}{\frac{t - t_i}{t_{i+1} - t_i} \; N_{i,1}(t)} \right]$$

$$+ \overset{II(i)}{\frac{t_{i+3} - t}{t_{i+3} - t_{i+1}}} \left[\overset{N(i+1),2}{\frac{t - t_{(i+1)}}{t_{(i+1)+2} - t_{(i+1)}} \; N_{(i+1),1}} + \frac{t_{(i+1)+2} - t}{t_{(i+1)+2} - t_{(i+1)+1}} \; N_{(i+1)+1,1}(t) \right]$$

$$\downarrow$$

$$+ \overset{II(i)}{\frac{t_{i+3} - t}{t_{i+3} - t_{i+1}}} \left[\overset{N(i+1),2}{\frac{t - t_{(i+1)}}{t_{i+3} - t_{(i+1)}} \; N_{(i+1),1}} + \frac{t_{i+3} - t}{t_{i+3} - t_{i+2}} \; N_{i+2,1}(t) \right]$$

****

$$N_{i,3}(t) = \overset{I(i)}{\frac{t - t_i}{t_{i+2} - t_i}} \left[\overset{N\,i,2}{\frac{t - t_i}{t_{i+1} - t_i} \; N_{i,1}(t)} \right]$$

****

If K=4 then sumover(i:j-(K-1),j)=sumover(i:j-3,j)

j s.t. $t_j <= t < t_{j+1}$

therefore $N_{i-3,4}(t)$; $N_{i-2,4}(t)$; $N_{i-1,4}(t)$; $N_{i,4}(t)$; are not equal to 0.0

•

Note: i=i-3 in the general defintion.

$$N_{i-3,4}(t) = \frac{t - t_{(i-3)}}{t_{(i-3)+3} - t_{(i-3)}} N_{(i-3),3}(t) + \frac{t_{(i-3)+4} - t}{t_{(i-3)+4} - t_{(i-3)+1}} N_{(i-3)+1,3}(t) .$$

I(i-3) **N(i-3),3** **II(i-3)** **N(i-2),3**

$$= \frac{t - t_{(i-3)}}{t_i - t_{(i-3)}} N_{(i-3),3}(t) + \frac{t_{i+1} - t}{t_{i+1} - t_{i-2}} N_{i-2,3}(t)$$

I(i-3) **N(i-3),3**

$$= \frac{t - t_{(i-3)}}{t_i - t_{(i-3)}} \left[\frac{t - t_{(i-3)}}{t_{i+1} - t_{(i-3)}} N_{(i-3),2}(t) + \frac{t_{(i-3)+3} - t}{t_{(i-3)+3} - t_{(i-3)+1}} N_{(i-3)+1,2}(t) \right]$$

N(i-3),3 N(i-3),3

I(i-3) I(i-3)a:N(i-3),2 I(i-3)b:N(i-),2

$$= \frac{t - t_{(i-3)}}{t_i - t_{(i-3)}} \left[\frac{t - t_{(i-3)}}{t_{i-1} - t_{(i-3)}} N_{(i-3),2}(t) + \frac{t_i - t}{t_i - t_{i-2}} N_{i-2,2}(t) \right]$$

N(i-3),3

I(i-3) I(i-3)a N(i-3),2

$$\frac{t - t_{(i-3)}}{t_i - t_{(i-3)}} \left[\frac{t - t_{(i-3)}}{t_{i-1} - t_{(i-3)}} \left[\frac{t - t_{(i-3)}}{t_{(i-3)+1} - t_{(i-3)}} N_{(i-3),1}(t) + \right. \right.$$

$$\left. \left. \frac{t_{(i-3)+2} - t}{t_{(i-3)+2} - t_{(i-3)+1}} N_{(i-3)+1,1}(t) \right] \right] +$$

↓

N(i-3),3

I(i-3) **I(i-3)a** **N(i-3),2**

$$\frac{t - t_{(i-3)}}{t_i - t_{(i-3)}} \left[\frac{t - t_{(i-3)}}{t_{i-1} - t_{(i-3)}} \left[\frac{t - t_{(i-3)}}{t_{i-2} - t_{(i-3)}} N_{(i-3),1}(t) + \frac{t_{i-1} - t}{t_{i-1} - t_{i-2}} N_{i-2,1}(t) \right] \right]$$

N(i-3),3

I(i-3) **I(i-3)b:N(i-2),2**

$$\frac{t - t_{(i-3)}}{t_i - t_{(i-3)}} \left[\frac{t_i - t}{t_i - t_{i-2}} N_{i-2,2}(t) \right]$$

N(i-3),3

I(i-3) **I(i-3)b** **N(i-2),2**

$$\frac{t - t_{(i-3)}}{t_i - t_{(i-3)}} \left[\frac{t_i - t}{t_i - t_{i-2}} \left[\frac{t - t_{(i-2)}}{t_{(i-2)+1} - t_{(i-2)}} N_{(i-2),1}(t) + \frac{t_{(i-2)+2} - t}{t_{(i-2)+2} - t_{(i-2)+1}} N_{(i-2)+1,1}(t) \right] \right]$$

↓

N(i-3),3

I(i-3) **I(i-3)b** **N(i-2),2**

$$\frac{t - t_{(i-3)}}{t_i - t_{(i-3)}} \left[\frac{t_i - t}{t_i - t_{i-2}} \left[\frac{t - t_{(i-2)}}{t_{i-1} - t_{(i-2)}} N_{(i-2),1}(t) + \frac{t_i - t}{t_i - t_{i-1}} N_{i-1,1}(t) \right] \right]$$

****

$$N_{i-3,4}(t) = \overset{\Pi(i-3)}{\frac{t_{i+1} - t}{t_{i+1} - t_{i-2}}} \left[\overset{N(i-2),3}{\frac{t_{i+1} - t}{t_{i+1} - t_{i-1}}} \left[\frac{t_{i+1} - t}{t_{i+1} - t_{i}} N_{i,1}(t) \right] \right]$$

****

Note: i=i-2 in the general definition

$$N_{i-2,4}(t) = \frac{t - t_{(i-2)}}{t_{(i-2)+3} - t_{(i-2)}} N_{(i-2),3} + \frac{t_{(i-2)+4} - t}{t_{(i-2)+4} - t_{(i-2)+1}} N_{(i-2)+1,3}(t)$$

I(i-2):N(i-2),3 **II(i-2):N(i-1),3**

$$= \frac{t - t_{(i-2)}}{t_{i+1} - t_{(i-2)}} N_{(i-2),3} + \frac{t_{i+2} - t}{t_{i+2} - t_{i-1}} N_{i-1,3}(t)$$

****

I(i-2) **N(i-2),3**

$$\mathbf{N}_{i-2,4}(t) = \frac{t - t_{(i-2)}}{t_{i+1} - t_{(i-2)}} \left[\frac{t_{i+1} - t}{t_{i+1} - t_{i-1}} \left[\frac{t_{i+1} - t}{t_{i+1} - t_i} N_{i,1}(t) \right] \right] +$$

II(i-2) **N(i-1),3a**

$$\frac{t_{i+2} - t}{t_{i+2} - t_{i-1}} \left[\frac{t - t_{(i-1)}}{t_{i+1} - t_{(i-1)}} \left[\frac{t_{i+1} - t}{t_{i+1} - t_i} N_{i,1}(t) \right] \right] +$$

II(i-2) **N(i-1),3b**

$$\frac{t_{i+2} - t}{t_{i+2} - t_{i-1}} \left[\frac{t_{i+2} - t}{t_{i+2} - t_i} \left[\frac{t - t_i}{t_{i+1} - t_i} N_{i,1}(t) \right] \right]$$

****

Note: i=i-1 in the general definition

$$N_{i-1,4}(t) = \frac{t - t_{(i-1)}}{t_{(i-1)+3} - t_{(i-1)}} N_{(i-1),3}(t) + \frac{t_{(i-1)+4} - t}{t_{(i-1)+4} - t_{(i-1)+1}} N_{(i-1)+1,3}(t)$$

$$= \overbrace{\frac{t - t_{(i-1)}}{t_{i+2} - t_{(i-1)}} N_{(i-1),3}(t)}^{\mathbf{I(i-1):N(i-1),3}} + \overbrace{\frac{t_{i+3} - t}{t_{i+3} - t_i} N_{i,3}(t)}^{\mathbf{II(i-1):N\ i,3}}$$

****

$$\mathbf{N_{i-1,4}(t)} = \overbrace{\frac{t - t_{(i-1)}}{t_{i+2} - t_{(i-1)}}}^{\mathbf{I(i-1)}} \left[\overbrace{\frac{t - t_{(i-1)}}{t_{i+1} - t_{(i-1)}}}^{\mathbf{N(i-1),3a}} \left[\frac{t_{i+1} - t}{t_{i+1} - t_i} N_{i,1}(t) \right] \right] +$$

$$\overbrace{\frac{t - t_{(i-1)}}{t_{i+2} - t_{(i-1)}}}^{\mathbf{I(i-1)}} \left[\overbrace{\frac{t_{i+2} - t}{t_{i+2} - t_i}}^{\mathbf{N(i-1),3b}} \left[\frac{t - t_i}{t_{i+1} - t_i} N_{i,1}(t) \right] \right] +$$

$$\overbrace{\frac{t_{i+3} - t}{t_{i+3} - t_i}}^{\mathbf{II(i-1)}} \left[\overbrace{\frac{t - t_i}{t_{i+2} - t_i}}^{\mathbf{N\ i,3}} \left[\frac{t - t_i}{t_{i+1} - t_i} N_{i,1}(t) \right] \right]$$

****

Note: i=i in the general definition

$$N_{i,4}(t) = \overset{\text{I(i):N i,3}}{\frac{t - t_i}{t_{i+3} - t_i}} N_{i,3}(t) + \overset{\text{II(i):N(i+1),3}}{\frac{t_{i+4} - t}{t_{i+4} - t_{i+1}}} N_{i+1,3}(t)$$

$$\overset{\text{II(i)}}{\frac{t_{i+4} - t}{t_{i+4} - t_{i+1}}} \left[\overset{\text{N(i+1),3}}{\frac{t - t_{(i+1)}}{t_{(i+1)+3} - t_{(i+1)}}} N_{(i+1),2}(t) + \frac{t_{(i+1)+3} - t}{t_{(i+1)+3} - t_{(i+1)+1}} N_{(i+1)+1}(t) \right]$$

$$\downarrow$$

$$= \overset{\text{II(i)}}{\frac{t_{i+4} - t}{t_{i+4} - t_{i+1}}} \left[\overset{\substack{\text{N(i+1),3} \\ \text{I(i+1)a:N(i+1),2}}}{\frac{t - t_{(i+1)}}{t_{i+4} - t_{(i+1)}}} N_{(i+1),2}(t) + \overset{\substack{\text{N(i+1),3} \\ \text{II(i+1)b:N(i+2),2}}}{\frac{t_{i+4} - t}{t_{i+4} - t_{i+2}}} N_{i+2,2}(t) \right]$$

$$\overset{\text{II(i)}}{\frac{t_{i+4} - t}{t_{i+4} - t_{i+1}}} \left[\overset{\substack{\text{N(i+1),3} \\ \text{I(i+1)a}}}{\frac{t - t_{(i+1)}}{t_{i+4} - t_{(i+1)}}} \left[\overset{\text{N(i+1),2}}{\frac{t - t_{(i+1)}}{t_{(i+1)+1} - t_{(i+1)}}} N_{(i+1),1}(t) + \frac{t_{(i+1)+2} - t}{t_{(i+1)+2} - t_{(i+1)+1}} N_{(i+1)+1,1}(t) \right] \right]$$

$$\downarrow$$

++

$$\overset{\text{II(i)}}{\frac{t_{i+4} - t}{t_{i+4} - t_{i+1}}} \left[\overset{\substack{\text{N(i+1),3} \\ \text{I(i+1)a}}}{\frac{t - t_{(i+1)}}{t_{i+4} - t_{(i+1)}}} \left[\frac{t - t_{(i+1)}}{t_{i+2} - t_{(i+1)}} N_{(i+1),1}(t) + \frac{t_{i+3} - t}{t_{i+3} - t_{i+2}} N_{i+2,1}(t) \right] \right]$$

++

N(i+1),3

II(i) II(i+1)b:N(i+2),2

$$\frac{t_{i+4} - t}{t_{i+4} - t_{i+1}} \left[\frac{t_{i+4} - t}{t_{i+4} - t_{i+2}} N_{i+2,2}(t) \right]$$

N(i+1),3

II(i) II(i+1)b N(i+2),2

$$\frac{t_{i+4} - t}{t_{i+4} - t_{i+1}} \left[\frac{t_{i+4} - t}{t_{i+4} - t_{i+2}} \left[\frac{t - t_{(i+2)}}{t_{(i+2)+1} - t_{(i+2)}} N_{(i+2),1} + \frac{t_{(i+2)+2} - t}{t_{(i+2)+2} - t_{(i+2)+1}} N_{(i+2)+1,1}(t) \right] \right]$$

|
V

++

N(i+1),3

II(i) II(i+1)b N(i+2),2

$$\frac{t_{i+4} - t}{t_{i+4} - t_{i+1}} \left[\frac{t_{i+4} - t}{t_{i+4} - t_{i+2}} \left[\frac{t - t_{(i+2)}}{t_{i+3} - t_{(i+2)}} N_{(i+2),1} + \frac{t_{i+4} - t}{t_{i+4} - t_{i+3}} N_{i+3,1}(t) \right] \right]$$

++

****

I(i) N i,3

$$N_{i,4}(t) = \frac{t - t_i}{t_{i+3} - t_i} \left[\frac{t - t_i}{t_{i+2} - t_i} \left[\frac{t - t_i}{t_{i+1} - t_i} N_{i,1}(t) \right] \right]$$

****

SUMMARY OF BASIS FUNCTIONS

K=1

****

$N_{i,1}(t) = 1.$

****

K=2

****

$$N_{i-1,2}(t) = \frac{t_{i+1} - t}{t_{i+1} - t_i} N_{i,1}(t)$$

$$N_{i,2}(t) = \frac{t - t_i}{t_{i+1} - t_i} N_{i,1}(t)$$

****

K=3

****

$$N_{i-2,3}(t) = \overset{II(i-2)}{\frac{t_{i+1} - t}{t_{i+1} - t_{i-1}}} \left[\overset{N(i-1),2}{\frac{t_{i+1} - t}{t_{i+1} - t_i} N_{i,1}(t)} \right]$$

$$N_{i-1,3}(t) = \overset{I(i-1)}{\frac{t - t_{(i-1)}}{t_{i+1} - t_{(i-1)}}} \left[\overset{N(i-1),2}{\frac{t_{i+1} - t}{t_{i+1} - t_i} N_{i,1}(t)} \right] + \overset{II(i-1)}{\frac{t_{i+2} - t}{t_{i+2} - t_i}} \left[\overset{N\ i,2}{\frac{t - t_i}{t_{i+1} - t_i} N_{i,1}(t)} \right]$$

$$N_{i,3}(t) = \overset{I(i)}{\frac{t - t_i}{t_{i+2} - t_i}} \left[\overset{N\ i,2}{\frac{t - t_i}{t_{i+1} - t_i} N_{i,1}(t)} \right]$$

****

K=4

****

$$N_{i-3,4}(t) = \overset{\text{II}(i-3)}{\frac{t_{i+1} - t}{t_{i+1} - t_{i-2}}} \left[\overset{N(i-2),3}{\frac{t_{i+1} - t}{t_{i+1} - t_{i-1}}} \left[\frac{t_{i+1} - t}{t_{i+1} - t_i} N_{i,1}(t) \right] \right]$$

$$N_{i-2,4}(t) = \overset{\text{I}(i-2)}{\frac{t - t_{(i-2)}}{t_{i+1} - t_{(i-2)}}} \left[\overset{N(i-2),3}{\frac{t_{i+1} - t}{t_{i+1} - t_{i-1}}} \left[\frac{t_{i+1} - t}{t_{i+1} - t_i} N_{i,1}(t) \right] \right] +$$

$$\overset{\text{II}(i-2)}{\frac{t_{i+2} - t}{t_{i+2} - t_{i-1}}} \left[\overset{N(i-1),3a}{\frac{t - t_{(i-1)}}{t_{i+1} - t_{(i-1)}}} \left[\frac{t_{i+1} - t}{t_{i+1} - t_i} N_{i,1}(t) \right] \right] +$$

$$\overset{\text{II}(i-2)}{\frac{t_{i+2} - t}{t_{i+2} - t_{i-1}}} \left[\overset{N(i-1),3b}{\frac{t_{i+2} - t}{t_{i+2} - t_i}} \left[\frac{t - t_i}{t_{i+1} - t_i} N_{i,1}(t) \right] \right]$$

$$N_{i-1,4}(t) = \overset{\text{I}(i-1)}{\frac{t - t_{(i-1)}}{t_{i+2} - t_{(i-1)}}} \left[\overset{N(i-1),3a}{\frac{t - t_{(i-1)}}{t_{i+1} - t_{(i-1)}}} \left[\frac{t_{i+1} - t}{t_{i+1} - t_i} N_{i,1}(t) \right] \right] +$$

$$\overset{\text{i}(i-1)}{\frac{t - t_{(i-1)}}{t_{i+2} - t_{(i-1)}}} \left[\overset{N(i-1),3b}{\frac{t_{i+2} - t}{t_{i+2} - t_i}} \left[\frac{t - t_i}{t_{i+1} - t_i} N_{i,1}(t) \right] \right] +$$

$$\overset{\text{II}(i-1)}{\frac{t_{i+3} - t}{t_{i+3} - t_i}} \left[\overset{N\ i,3}{\frac{t - t_i}{t_{i+2} - t_i}} \left[\frac{t - t_i}{t_{i+1} - t_i} N_{i,1}(t) \right] \right]$$

$$N_{i,4}(t) = \overset{\text{I}(i)}{\frac{t - t_i}{t_{i+3} - t_i}} \left[\overset{N\ i,3}{\frac{t - t_i}{t_{i+2} - t_i}} \left[\frac{t - t_i}{t_{i+1} - t_i} N_{i,1}(t) \right] \right]$$

****

APPENDIX B
Subroutine Index and BSPLINE Program Calltree

SUBROUTINE INDEX FOR THE PROGRAM BSPLINE

Note: **BOLD-FACE** subroutines were written by the author.

ACTMDB: subroutine used to turn on the active mesh in MOVING POINTER in the MODIFY BSPLINE module of the program.

AIJVAL: subroutine to evaluate the reduced set of vertices needed in the evaluation of derivatives and partials of the B-spline basis functions.

BSPDFM: subroutine that acts as the entry point and control structure for the program module DISPLAY FILE MANAGER.

BSPGNB: subroutine that acts as the entry point and control structure for the program module GENERATE BSPLINE.

BSPLINE: main program used to access the other modules

BSPMBD: subroutine that acts as the entry point and control structure for the program module MODIFY BSPLINE.

CHGPNT: subroutine used to graphically change the location of control points in the command MOVE PIJ"S in the module MODIFY BSPLINE.

CMCLOS: PLOT 10 subroutine to end graphic communication and empty graphics buffer.

CMDSTR: subroutine used to parse the command line for user input.

CMOPEN: PLOT 10 subroutine to open graphics communication with terminal.

CMSCMD: subroutine to execute system commands from FORTRAN programs.

CRTDFM: subroutine used to execute the CREATE FILE command in the DISPLAY FILE MANAGER module, creates display arrays from the existing program arrays if they exist.

DRAW: PLOT 10 subroutine to draw a from present location to location specified.

DRWMSH: subroutine to execute DRAW MESH commands in draw entire control mesh in GENERATE BSPLINE and DISPLAY FILE MANAGER modules of the program.

DRWPTR: subroutine to draw the surface pointer in the MODIFY BSPLINE module.

ERSDFM: subroutine used to execute command ERASE in the module DISPLAY FILE MANAGER erases the screen.

FRAME: subroutine to draw the standard presentation window.

FRAMER: subroutine to draw the full screen presentation window in the module DISPLAY FILE MANAGER.

GETDAT: subroutine used to get the counters used in calculating the points on the streamlines.

GETFIL: subroutine used to get filename and filetype when inputting and outputting files to and from the program

GETPIJ: subroutine that returns the starting and ending control points needed to generate the given parameter.

GETPNT: subroutine used by the command MOVE PIJ"S in the module MODIFY BSPLINE to get control point the user wants to modify.

GETSNS: subroutine used to get the sensitivity used in moving the points in MOVE PIJ"S command in the module MODIFY BSPLINE.

GRSTOP: PLOT 10 subroutine to terminate the graphics.

INTDFM: subroutine used to execute the command RESET in the module DISPLAY FILE MANAGER, reintializes the display arrays.

INTSPL: subroutine used to execute the command RESET in the module GENERATE BSPLINE, reinitializes the database and streamlines and normal arrays.

ISOSPL: subroutine used to execute the command ISOMETRIC in the module GENERATE BSPLINE, generates an isometric of the present data set by rotating first 45 degrees about the y axis and then 35.246 degrees about the x axis.

KAS2AM: PLOT 10 subroutine used to send commands to the terminal from the host.

LBLMSH: subroutine used by MOVE PIJ"S in the module MODIFY BSPLINE, to list the labels of the active mesh in the margin of the drawing area.

MDZOOM: subroutine used to do a soft clip of the drawing area or window.

MENLST: subroutine used to draw the menu in the presentation window.

MNMXER: subroutine used to find the 3 dimensional minimums and maximums of the specified control mesh which are then used to size the drawing window.

MOVE: PLOT 10 subroutine to move the graphics cursor without drawing.

MPJMDB: subroutine used to implement the command MOVE PIJ"S in the module MODIFY BSPLINE.

MVPMDB: subroutine to implement the MOVING POINTER command in the module MODIFY BSPLINE.

NEWPAG: PLOT 10 subroutine to clear the screen.

NIKVAL: subroutine to evaluate the B-spline basis function.

NMLGNB: subroutine to execute the command DRAW NORMALS in the module GENERATE BSPLINE, draws the normals to the surface.

NRMVCT: subroutine to calculate the normal vector given the point on the surface and the parametric partial derivatives.

NUMBER: subroutine to convert a character string from the command line into a real or integer number with error checking.

PRTFIL: subroutine to execute the command PRINT FILE which will print all the program variables to the screen or to a file.

RDRDFM: subroutine to execute the command READ FILE in the module DISPLAY FILE MANAGER, inputs a display file into the display arrays.

RDRSPL: subroutine to execute the command READ FILE in the module GENERATE BSPLINE, inputs a data file to the program.

ROTPTS: subroutine to rotate an array of points using a homogenous transformation.

ROTSPL: subroutine to execute the command ROTATE in the module GENERATE BSPLINE, rotates the data file by a specified angle and about a specified axis.

SAVDFM: subroutine to execute the command SAVE FILE in the module DISPLAY FILE MANAGER, outputs a display file to a specified file on the disk.

SAVSPL: subroutine to execute SAVE command in the module GENERATE BSPLINE, outputs a data file to a specified file on the disk.

SCLDFM: subroutine to execute SCALE OUTPUT command in the module DISPLAY FILE MANAGER, scales the size of the full screen output window by a factor between 0.01 and 1.

SCLPTS: subroutine to scale an array of points using a homogenous scaling transformation.

SCLSPL: subroutine to execute the command SCALE DATA in the module GENERATE BSPLINE, scales the data file by a specified amount.

SCNDFM: subroutine to execute the command SCREEN TYPE in the module DISPLAY FILE MANAGER, used to specify whether the standard presentation window or full screen presentation window is used in the module.

SETSCN: subroutine used to set up the window and viewport by finding the minimums and maximums of the control mesh and specifying a hard clipped window.

SETUPS: subroutine to put the cursor key functions in the presentation window in the MODIFY BSPLINE module.

SPLPRT: subroutine to evaluate the parametric partial derivatives used to calculate the surface normals.

SPLSFC: subroutine to evaluate the point on a surface.

STRDFM: subroutine to execute the command DRAW STREAMLINES in the module DISPLAY FILE MANAGER, draws the surface streamlines in the display file.

STRGNB: subroutine to execute the command DRAW STREAMLINES in the module GENERATE BSPLINE, draws the streamlines generated using the STREAMLINES AND NORMALS command.

STRMDB: subroutine to execute the command DRAW STREAMLINES in the module MODIFY BSPLINE, draw the streamlines generated when modifying the surface.

STRNML: subroutine to execute the command STREAMLINES AND NORMALS in the module GENERATE BSPLINE, evaluates the points and normals to the surface used in display.

TEXT: PLOT 10 subroutine to put text on the graphics screen.

TRMCHR: subroutine to execute commands to the terminal with the command specified by a character string.

TRMCMD: subroutine to execute commands to the terminal with the command specified by an integer number, replaced by TRMCHR.

TRML1: subroutine to change the terminal from 4014 graphics mode to ANSI mode.

TRML2: subroutine to change the terminal from ANSI mode to 4014 graphics mode.

TRML3: subroutine to clear the entire ANSI screen.

TRML4: subroutine to position the ANSI cursor at a specified location on the screen.

TRML5: subroutine to clear the graphics screen.

TRML7: subroutine to erase the ANSI screen above the current cursor location.

TRML8: subroutine to change the graphics drawing cursor to the "CLEAR" submode.

TRML9: subroutine to change the graphics drawing cursor to the "XOR" submode.

TRML10: subroutine to change the graphics drawing cursor to the "OR" submode.

TRML11: subroutine to set the ANSI scrolling region to a specified area.

TRML12: subroutine to erase the ANSI screen below the current cursor location.

TRML13: subroutine to set the ANSI relative origin mode.

TRML14: subroutine used to set the ANSI absolute origin mode.

TRML15: subroutine used to set the graphics line type in the standard submode.

TRML16: subroutine used to set the graphics line type in the write through submode.

TXICUR: PLOT 10 subroutine used to position graphics text.

TXSIZE: PLOT 10 subroutine used to specify the size of graphics text.

UPDMDB: subroutine used to execute the UPDATE command in the module MODIFY BSPLINE, recalculates the streamlines and normals after changes have been made in MOVE PIJ"S.

VWPORT: PLOT 10 subroutine used to specify the portion of the screen to be used for the window.

WINDOW: PLOT 10 subroutine used to specify the dimension of the viewport.

CALLTREE FOR BSPLINE PROGRAM

UTILITIES

System Utilities

CMSCMD(CMDCMS,IRC)

PLOT 10 Subroutines

GRSTRT(IDEVIC,IOPT)
GRSTOP
CMOPEN
CMCLOS
NEWPAG
VWPORT(XMIN,XMAX,YMIN,YMAX)
WINDOW(XMIN,XMAX,YMIN,YMAX)
MOVE(PX,PY)
DRAW(PX,PY)
TEXT(ILENST,ICHRAY)
TXSIZE(ISIZE,PXSIZE,PYSIZE)
TXICUR(IPOS)
KAS2AM(ICNT,IASRAY,IAMRAY)

Terminal Utilities

TRMCHR(CHRCMD)
TRMCMD(NCMND)
    TRML1
        Plot10: CMOPEN,KAS2AM
    TRML2
        Plot10: CMOPEN,KAS2AM
    TRML3
        Plot10: CMOPEN,KAS2AM
    TRML4(LNE1,LNE2,LNE3,LNE3)
        Plot10: CMOPEN,KAS2AM
    TRML5
        Plot10: CMOPEN,KAS2AM
    TRML7
        Plot10: CMOPEN,KAS2AM
    TRML8
        Plot10: CMOPEN,KAS2AM
    TRML9
        Plot10: CMOPEN,KAS2AM
    TRML10
        Plot10: CMOPEN,KAS2AM
    TRML11(TOP1,TOP2,BOT1,BOT2)
        Plot10: CMOPEN,KAS2AM
    TRML13
        Plot10: CMOPEN,KAS2AM
    TRML14
        Plot10: CMOPEN,KAS2AM
    TRML15(NUMBER)

Plot10: CMOPEN,KAS2AM
TRML16(NUMBER)
Plot10: CMOPEN,KAS2AM

Screen Utilities

SETSCN(PTS,NIP,NJP,IDMPTS,SCRN,VWSCL)
Plot 10: CMOPEN,CMCLOS,VWPORT,WINDOW
MNMXER(PTS,ISTART,ISTOP,JSTART,JSTOP,IDMPTS,
XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX)
MDZOOM(XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX,SCREEN,SCL,BORDER)
Plot 10: CMOPEN,CMCLOS,VWPORT,WINDOW
MENLST
Plot 10: CMOPEN,CMCLOS,TXSIZE,TEXT,MOVE,DRAW,WINDOW,VWPORT
FRAME
Plot 10: CMOPEN,CMCLOS,NEWPAG,MOVE,DRAW,WINDOW,VWPORT
Terminal: **TRMCMD**
FRAMER(SCN,SCL)
Plot 10: CMOPEN,CMCLOS,NEWPAG,MOVE,DRAW,WINDOW,VWPORT
Terminal: **TRMCMD**

Command Line Utilities

CMDSTR(IPCMND,LCM,CMSTR,ERR)
NUMBER(NTYPE,NSTR,I,R,NERR)
GETFIL(FNAME,FTYPE,TRYAGN)
Command: **CMDSTR**

Matrix Utilities

ROTPTS(PTS,NIP,NJP,IDMPTS,ANGLE,AXIS,RPTS)
SCLPTS(PTS,NIP,NJP,IDMPTS,SCLFCT,SPTS)

BSPLINE **MAIN**

Plot 10: GRSTRT,GRSTOP,CMOPEN,CMCLOS
Terminal: **TRMCHR**
Command: **CMDSTR**
Screen: **FRAME**
Others: **INTSPL,INTDFM** will be specified in GENERATE BSPLINE and DISPLAY FILE MANAGER

<GNB> GENERATE BSPLINE: **BSPGNB**

<DFM> DISPLAY FILE MANAGER: **BSPDFM**

<PF> PRINT FILE: **PRTFIL**

System: CMSCMD
Plot 10: CMOPEN,CMCLOS,NEWPAG
Command: **CMDSTR**
Screen: **FRAME,FRAMER**

<HP> HELP: **HLPCMD** NOTE: command not implemented

<QT> QUIT PROGRAM

GENERATE BSPLINE: **BSPGNB**
Plot 10: CMOPEN,CMCLOS
Terminal: **TRMCHR**
Command: **CMDSTR**
Screen: **SETSCN,FRAME**

<RF> READ FILE **RDRSPL**
System: CMSCMD
Plot 10: CMOPEN,CMCLOS
Terminal: **TRMCMD**
Command: **CMDSTR,GETFIL**
Others: **INTSPL**

<DM> DRAW MESH: **DRWMSH**(PTS,NIP,NJP,IDMPTS)
Plot 10: CMOPEN,CMCLOS,MOVE,DRAW
Terminal: **TRMCMD**

<DN> DRAW NORMALS: **NMLGNB**
Plot 10: CMOPEN,CMCLOS,MOVE,DRAW
Terminal: **TRMCMD**

<DS> DRAW STREAMLINES: **STRGNB**
Plot 10: CMOPEN,CMCLOS,MOVE,DRAW
Terminal: **TRMCMD**

<IS> ISOMETRIC: **ISOSPL**
Plot 10: CMOPEN,CMCLOS
Terminal: **TRMCMD**
Screen: **FRAME,SETSCN**
Matrix: **ROTPTS**
Others: **DRWMSH**(PTS,NIP,NJP,IDMPTS)

<RT> ROTATE: **ROTSPL**
Plot 10: CMOPEN,CMCLOS
Terminal: **TRMCMD**
Command: **CMDSTR**
Screen: **FRAME,SETSCN**
Matrix: **ROTPTS**
Others: **DRWMSH**(PTS,NIP,NJP,IDMPTS)

<SD> SCALE DATA: **SCLPTS**
Plot 10: CMOPEN,CMCLOS
Terminal: **TRMCMD**
Command: **CMDSTR,NUMBER**
Screen: **FRAME,SETSCN**
Matrix: **SCLPTS**
Others: **DRWMSH**(PTS,NIP,NJP,IDMPTS)

<SN> STREAMLINES AND NORMALS: **STRNML**
Plot 10: CMOPEN,CMCLOS
Others:
GETDAT(KNTIMX,KNTJMX,INTVAL,MXSTRM,MXPNTS,
ISTRM,IPNTS,JSTRM,JPNTS)

```
      Plot 10: CMOPEN,CMCLOS
      Terminal: TRMCMD
      Command: CMDSTR,NUMBER
SPLSFC(UI,VJ,PIJ,IORDER,KNOTI,INUM,JORDER,KNOTJ,JNUM,MXNUM,IDIM,S)
      NIKVAL(T,KORDER,IKNOT,KNOT,KNTMAX,JDRV,NIK)
SPLPRT(UI,VJ,PIJ,IORJ,IORDER,KNOTI,INUM,
                    JORDER,KNOTJ,JNUM,MXNUM,NDRV,IDIM,SP)
             AIJVAL(KORDER,IKNOT,KNOT,AIJVRT,IDIM,JDRV, IJ)
      NIKVAL(T,KORDER,IKNOT,KNOT,KNTMAX,JDRV,NIK)
NRMVCT(UVPNT,PU,PV,IDIM,VCTNML)
```

<MDB> MODIFY BSPLINE: **BSPMDB**

<PF> PRINT FILE: **PRTFIL** same as in BSPLINE MAIN

<ER> ERASE
Screen: **FRAME,SETSCN**

<RS> RESET: **INTSPL**
Plot 10: CMOPEN,CMCLOS
Terminal: **TRMCMD**

<SV> SAVE FILE: **SAVSPL**
System: CMSCMD
Plot 10: CMOPEN,CMCLOS
Terminal: **TRMCMD**
Command: **CMDSTR,GETFIL**

<QT> QUIT
Terminal: **TRMCHR**

```
MODIFY BSLPLINE: BSPMDB
      Terminal: TRMCHR
      Command: CMDSTR
      Screen: FRAME,SETSCN
      Others:
      SETUPS(NUMSET)
            Plot 10: CMOPEN,CMCLOS,TEXT,TXSIZE,WINDOW,VWPORT
            Terminal: TRMCHR

<MP> MOVING POINTER: MVPMDB(ISTART,ISTOP,DIVI,JSTART,JSTOP,DIVJ)
      Terminal: TRMCHR,TRMCMD
      Command: CMDSTR
      Moving Pointer Utilities
      GETPIJ(T,KNOT,KORDER,NVERT,BGNPIJ,ENDPIJ)
      DRWPTR(UVX,UVY,UVZ,NVX,NVY,NVZ)
            Plot 10: CMOPEN,CMCLOS,MOVE,DRAW
            Terminal: TRMCHR
      Others:
      MSHMDB(PTS,ISTART,ISTOP,JSTART,JSTOP,IDMPTS)
            Plot 10: CMOPEN,CMCLOS,MOVE,DRAW
            Terminal: TRMCHR

<DA> DRAW ACTIVE MESH: ACTMBD
      Terminal: TRMCHR
      Command: CMDSTR

<DM> DRAW MESH: MSHMDB(PTS,ISTART,ISTOP,JSTART,JSTOP,IDMPTS)
      Plot 10: CMOPEN,CMCLOS,MOVE,DRAW
      Terminal: TRMCHR

<DS> DRAW STREAMLINES: STRMDB(ISTRTS,ISTOPS,JSTRTS,JSTOPS,
                              ISTRTP,ISTOPP,JSTRTP,JSTOPP,IDIM)
      Plot 10: CMOPEN,CMCLOS,MOVE,DRAW
      Terminal: TRMCMD

<MV> MOVE PIJ"S: MPJMDB
      Plot 10: CMOPEN,CMCLOS
      Terminal: TRMCHR
      Command: CMDSTR
      Screen: FRAME,MNMXER,MDZOOM
      Move PIJ"S Utilities
      GETSNS(SNS)
            Terminal: TRMCHR
            Command: CMDSTR,NUMBER
      GETPNT(ISTART,ISTOP,JSTART,JSTOP,ILOC,JLOC)
            Terminal: TRMCHR
            Command: CMDSTR,NUMBER
      CHGPNT(PNTS,IPNT,JPNT,IDIM,ISTART,ISTOP,JSTART,JSTOP)
            Plot 10: CMOPEN,CMCLOS,MOVE,DRAW
            Terminal: TRMCHR
      LBLMSH(IBGN,IEND,JBGN,JEND)
            Plot 10: CMOPEN,CMCLOS,MOVE,DRAW,TEXT,TXSIZE,TXICUR
            Terminal: TRMCHR
      DRWPTR(UVX,UVY,UVZ,NVX,NVY,NVZ)
            Plot 10: CMOPEN,CMCLOS,MOVE,DRAW
```

```
      Terminal: TRMCHR
GETPIJ(T,KNOT,KORDER,NVERT,BGNPIJ,ENDPIJ)
MSHMBD(PTS,ISTART,ISTOP,JSTART,JSTOP,IDMPTS)
      Plot 10: CMOPEN,CMCLOS,MOVE,DRAW
      Terminal: TRMCHR
SETUPS(NUMSET)
      Plot 10: CMOPEN,CMCLOS,TEXT,TXSIZE,WINDOW,VWPORT
      Terminal: TRMCHR

<UP> UPDATE: UPDMBD(ISTRTS,ISTOPS,JSTRTS,JSTOPS,
                    ISTRTP,ISTOPP,JSTRTP,JSTOPP,
                    IDVSTR,IDVPTS,JDVSTR,JDVPTS)
      Plot 10: CMOPEN,CMCLOS
      Terminal: TRMCHR
      Others:
      SPLSFC(UI,VJ,PIJ,IORDER,KNOTI,INUM,JORDER,KNOTJ,JNUM,MXNUM,IDIM,S)
            NIKVAL(T,KORDER,IKNOT,KNOT,KNTMAX,JDRV,NIK)
      SPLPRT(UI,VJ,PIJ,IORJ,IORDER,KNOTI,INUM,
                        JORDER,KNOTJ,JNUM,MXNUM,NDRV,IDIM,SP)
            AIJVAL(KORDER,IKNOT,KNOT,AIJVRT,IDIM,JDRV,AIJ)
            NIKVAL(T,KORDER,IKNOT,KNOT,KNTMAX,JDRV,NIK)
      NRMVCT(UVPNT,PU,PV,IDIM,VCTNML)

<PF> PRINT FILE: same as in BSPLINE MAIN

<ER> ERASE
      Terminal: TRMCHR
      Screen: FRAME,SETSCN

<QT> QUIT
      Terminal: TRMCHR
```

DISPLAY FILE MANAGER: **BSPDFM**
Plot 10: CMOPEN,CMCLOS
Terminal: **TRMCHR,TRMCMD**
Command: **CMDSTR**
Screen: **FRAME,SETSCN**

<CR> CREATE FILE: **CRTDFM**
Plot 10: CMOPEN,CMCLOS
Terminal: **TRMCHR**

<RF> READ FILE: **RDRDFM**
System: CMSCMD
Plot 10: CMOPEN,CMCLOS
Terminal: **TRMCMD**
Command: **CMDSTR,GETFIL**
Other:
INTDFM
Plot 10: CMOPEN,CMCLOS

<DM> DRAW MESH: **DRWMSH**(PTS,NIP,NJP,IDMPTS)
Plot 10: CMOPEN,CMCLOS,MOVE,DRAW
Terminal: **TRMCMD**

<DS> DRAW STREAMLINES **STRDFM**
Plot 10: CMOPEN,CMCLOS,MOVE,DRAW
Terminal: **TRMCMD**

<ST> SCREEN TYPE: **SCNDFM**
Plot 10: CMOPEN,CMCLOS
Terminal: **TRMCMD**
Command: **CMDSTR**
Other:
ERSDFM:
Plot 10: CMOPEN,CMCLOS
Terminal: **TRMCMD**
Screen: **FRAME,FRAMER,SETSCN**

<SO> SCALE OUTPUT: **SCLDFM**
Plot 10: CMOPEN,CMCLOS
Terminal: **TRMCMD**
Command: **CMDSTR,NUMBER**
Other:
ERSDFM:
Plot 10: CMOPEN,CMCLOS
Terminal: **TRMCMD**
Screen: **FRAME,FRAMER,SETSCN**

<PF> PRINT FILE: same as in BSLPLINE MAIN

<ER> ERASE: **ERSDFM**
Plot 10: CMOPEN,CMCLOS
Terminal: **TRMCMD**
Screen: **FRAME,FRAMER,SETSCN**

<RS> RESET: **INTDFM**

Plot 10: CMOPEN,CMCLOS
Terminal:**TRMCMD**

<SV> SAVE: **SAVDFM**
System: CMSCMD
Plot 10: CMOPEN,CMCLOS
Terminal: **TRMCMD**
Command: **CMDSTR,GETFIL**

<QT> QUIT:
Terminal: **TRMCHR**

APPENDIX C

Bell Data File

a. PRINT FILE Output

b. Data file

c. Plot of I Parameter

d. Plot of J Parameter

```
*** DATA FROM THE SPLINE SURFACE FILE *** :BELL     DATA

ORDER KI:   3
NUMBER OF I VERTICES:  9  (ZERO INDEX:  8).
VERTICE DIMENSION:   4
KNTIMX:   4

*** KNTI VECTOR ***
  0        0
  1        0
  2        0
  3        1
  4        1
  5        2
  6        2
  7        3
  8        3
  9        4
 10        4
 11        4

ORDER KJ:   4
NUMBER OF J VERTICE:   9  (ZERO INDEX:  8)
VERTICE DIMENSION:   4
KNTMX2:   6

*** KNTJ VECTOR ***
  0        0
  1        0
  2        0
  3        0
  4        1
  5        2
  6        3
  7        4
  8        5
  9        6
 10        6
 11        6
 12        6

*** POINTS OF THE POLYGON MESH ***

                          XDIM     YDIM     ZDIM     HDIM
PIJ( I, J)
      0   0               0.01    11.20     0.00     1.00
      1   0               0.01    11.20     0.01     1.00
      2   0               0.00    22.40     0.02     2.00
      3   0              -0.01    11.20     0.01     1.00
      4   0              -0.01    11.20     0.00     1.00
```

5	0	-0.01	11.20	-0.01	1.00
6	0	0.00	22.40	-0.02	2.00
7	0	0.01	11.20	-0.01	1.00
8	0	0.01	11.20	0.00	1.00

PIJ(I, J)

0	1	1.00	11.00	0.00	1.00
1	1	1.00	11.00	1.00	1.00
2	1	0.00	22.00	2.00	2.00
3	1	-1.00	11.00	1.00	1.00
4	1	-1.00	11.00	0.00	1.00
5	1	-1.00	11.00	-1.00	1.00
6	1	0.00	22.00	-2.00	2.00
7	1	1.00	11.00	-1.00	1.00
8	1	1.00	11.00	0.00	1.00

PIJ(I, J)

0	2	2.00	10.50	0.00	1.00
1	2	2.00	10.50	2.00	1.00
2	2	0.00	21.00	4.00	2.00
3	2	-2.00	10.50	2.00	1.00
4	2	-2.00	10.50	0.00	1.00
5	2	-2.00	10.50	-2.00	1.00
6	2	0.00	21.00	-4.00	2.00
7	2	2.00	10.50	-2.00	1.00
8	2	2.00	10.50	0.00	1.00

PIJ(I, J)

0	3	3.00	9.50	0.00	1.00
1	3	3.00	9.50	3.00	1.00
2	3	0.00	19.00	6.00	2.00
3	3	-3.00	9.50	3.00	1.00
4	3	-3.00	9.50	0.00	1.00
5	3	-3.00	9.50	-3.00	1.00
6	3	0.00	19.00	-6.00	2.00
7	3	3.00	9.50	-3.00	1.00
8	3	3.00	9.50	0.00	1.00

PIJ(I, J)

0	4	4.00	6.50	0.00	1.00
1	4	4.00	6.50	4.00	1.00
2	4	0.00	13.00	8.00	2.00
3	4	-4.00	6.50	4.00	1.00
4	4	-4.00	6.50	0.00	1.00
5	4	-4.00	6.50	-4.00	1.00
6	4	0.00	13.00	-8.00	2.00
7	4	4.00	6.50	-4.00	1.00
8	4	4.00	6.50	0.00	1.00

PIJ(I, J)

0	5	5.00	3.00	0.00	1.00
1	5	5.00	3.00	5.00	1.00
2	5	0.00	6.00	10.00	2.00
3	5	-5.00	3.00	5.00	1.00
4	5	-5.00	3.00	0.00	1.00
5	5	-5.00	3.00	-5.00	1.00
6	5	0.00	6.00	-10.00	2.00
7	5	5.00	3.00	-5.00	1.00
8	5	5.00	3.00	0.00	1.00

PIJ(I, J)

0	6	6.00	2.00	0.00	1.00
1	6	6.00	2.00	6.00	1.00
2	6	0.00	4.00	12.00	2.00
3	6	-6.00	2.00	6.00	1.00
4	6	-6.00	2.00	0.00	1.00
5	6	-6.00	2.00	-6.00	1.00
6	6	0.00	4.00	-12.00	2.00
7	6	6.00	2.00	-6.00	1.00
8	6	6.00	2.00	0.00	1.00

PIJ(I, J)

0	7	7.50	1.50	0.00	1.00
1	7	7.50	1.50	7.50	1.00
2	7	0.00	3.00	15.00	2.00
3	7	-7.50	1.50	7.50	1.00
4	7	-7.50	1.50	0.00	1.00
5	7	-7.50	1.50	-7.50	1.00
6	7	0.00	3.00	-15.00	2.00
7	7	7.50	1.50	-7.50	1.00
8	7	7.50	1.50	0.00	1.00

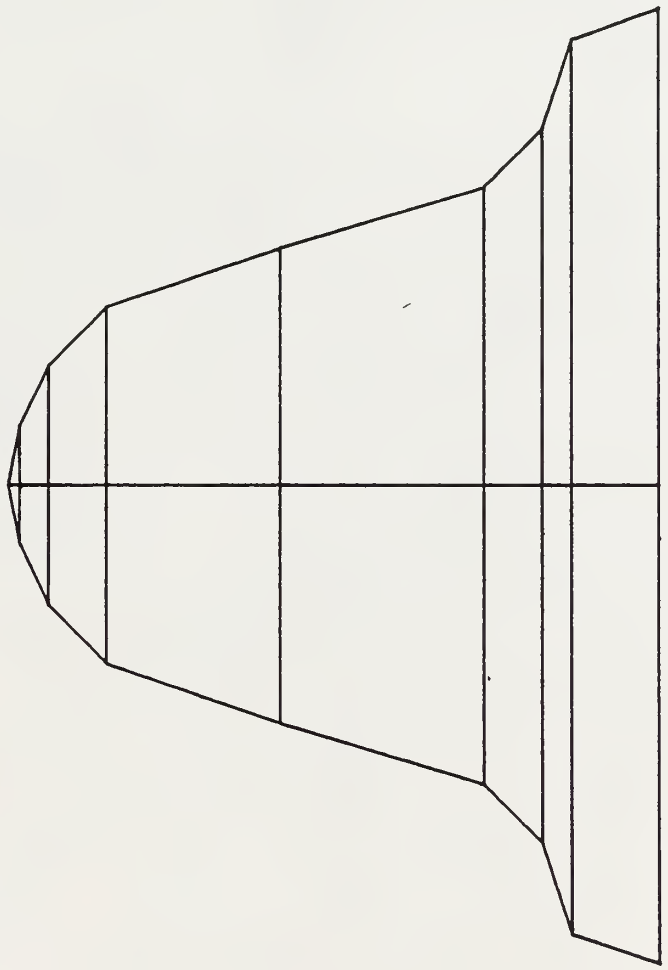
PIJ(I, J)

0	8	8.00	0.00	0.00	1.00
1	8	8.00	0.00	8.00	1.00
2	8	0.00	0.00	16.00	2.00
3	8	-8.00	0.00	8.00	1.00
4	8	-8.00	0.00	0.00	1.00
5	8	-8.00	0.00	-8.00	1.00
6	8	0.00	0.00	-16.00	2.00
7	8	8.00	0.00	-8.00	1.00
8	8	8.00	0.00	0.00	1.00

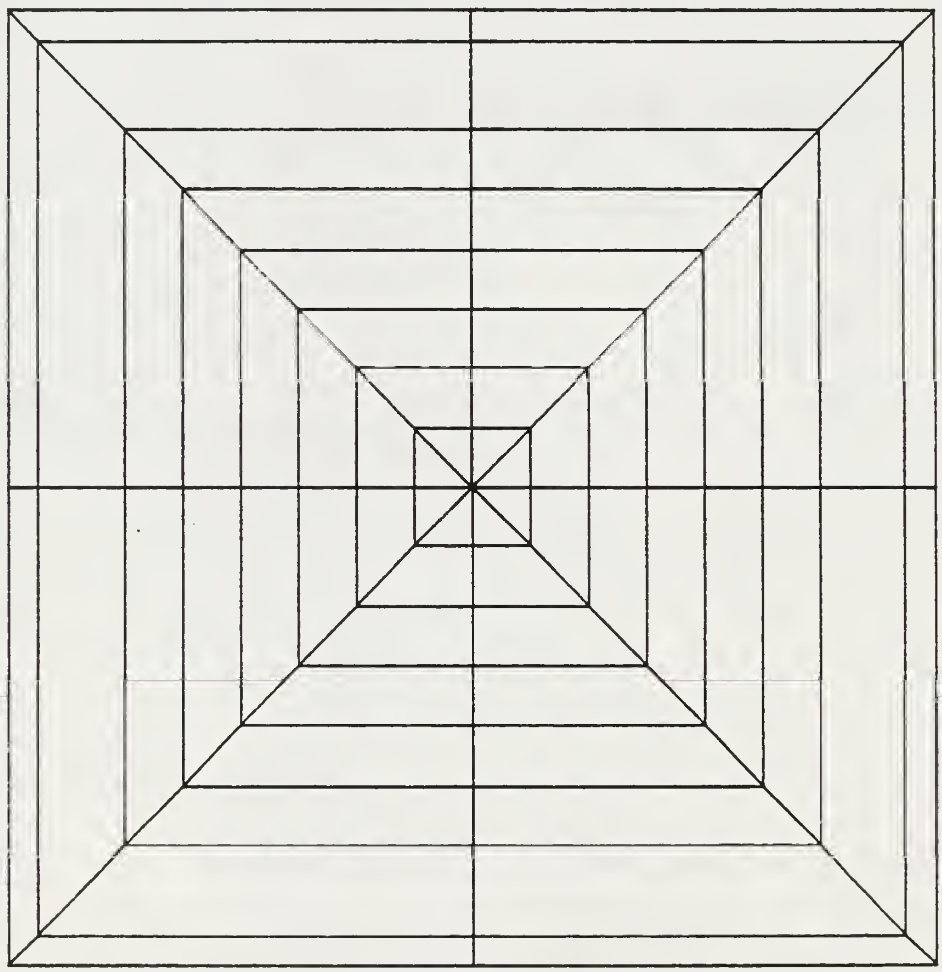
```
BSPLINE SURFACE FILE
  3   4   8   8   -
  0
  0
  0
  1
  1
  2
  2
  3
  3
  4
  4
  4
  0
  0
  0
  0
  1
  2
  3
  4
  5
  6
  6
  6
  6
   0.01   11.20    0.00    1.00
   0.01   11.20    0.01    1.00
   0.00   22.40    0.02    2.00
  -0.01   11.20    0.01    1.00
  -0.01   11.20    0.00    1.00
  -0.01   11.20   -0.01    1.00
   0.00   22.40   -0.02    2.00
   0.01   11.20   -0.01    1.00
   0.01   11.20    0.00    1.00
   1.00   11.00    0.00    1.00
   1.00   11.00    1.00    1.00
   0.00   22.00    2.00    2.00
  -1.00   11.00    1.00    1.00
  -1.00   11.00    0.00    1.00
  -1.00   11.00   -1.00    1.00
   0.00   22.00   -2.00    2.00
   1.00   11.00   -1.00    1.00
   1.00   11.00    0.00    1.00
   2.00   10.50    0.00    1.00
   2.00   10.50    2.00    1.00
   0.00   21.00    4.00    2.00
  -2.00   10.50    2.00    1.00
  -2.00   10.50    0.00    1.00
  -2.00   10.50   -2.00    1.00
   0.00   21.00   -4.00    2.00
   2.00   10.50   -2.00    1.00
   2.00   10.50    0.00    1.00
   3.00    9.50    0.00    1.00
```

3.00	9.50	3.00	1.00
0.00	19.00	6.00	2.00
-3.00	9.50	3.00	1.00
-3.00	9.50	0.00	1.00
-3.00	9.50	-3.00	1.00
0.00	19.00	-6.00	2.00
3.00	9.50	-3.00	1.00
3.00	9.50	0.00	1.00
4.00	6.50	0.00	1.00
4.00	6.50	4.00	1.00
0.00	13.00	8.00	2.00
-4.00	6.50	4.00	1.00
-4.00	6.50	0.00	1.00
-4.00	6.50	-4.00	1.00
0.00	13.00	-8.00	2.00
4.00	6.50	-4.00	1.00
4.00	6.50	0.00	1.00
5.00	3.00	0.00	1.00
5.00	3.00	5.00	1.00
0.00	6.00	10.00	2.00
-5.00	3.00	5.00	1.00
-5.00	3.00	0.00	1.00
-5.00	3.00	-5.00	1.00
0.00	6.00	-10.00	2.00
5.00	3.00	-5.00	1.00
5.00	3.00	0.00	1.00
6.00	2.00	0.00	1.00
6.00	2.00	6.00	1.00
0.00	4.00	12.00	2.00
-6.00	2.00	6.00	1.00
-6.00	2.00	0.00	1.00
-6.00	2.00	-6.00	1.00
0.00	4.00	-12.00	2.00
6.00	2.00	-6.00	1.00
6.00	2.00	0.00	1.00
7.50	1.50	0.00	1.00
7.50	1.50	7.50	1.00
0.00	3.00	15.00	2.00
-7.50	1.50	7.50	1.00
-7.50	1.50	0.00	1.00
-7.50	1.50	-7.50	1.00
0.00	3.00	-15.00	2.00
7.50	1.50	-7.500	1.00
7.50	1.50	0.00	1.00
8.00	0.00	0.00	1.00
8.00	0.00	8.00	1.00
0.00	0.00	16.00	2.00
-8.00	0.00	8.00	1.00
-8.00	0.00	0.00	1.00
-8.00	0.00	-8.00	1.00
0.00	0.00	-16.00	2.00
8.00	0.00	-8.00	1.00
8.00	0.00	0.00	1.00

APPENDIX C: display of bell data file for parameter J



APPENDIX C: bell data file for parameter I



VITA

Carl A. Kemner
candidate for the
Degree of Masters of Science

Thesis title: A B-spline Surface Modeler with Interactive Modification of the Control Mesh.

Major field: Mechanical Engineering

Biographical Data

Personal Data: Born in Quincy, Illinois June 21, 1952 the son of William and Jennie May Kemner.

Education: Graduated from Christian Brothers High School Quincy, Illinois in 1970; received Bachelor of Science in Physics and Bachelor of Science in Industrial Arts Education in 1977 from the University of Illinois; received Bachelor of Science in Mechanical Engineering in 1984 from Kansas State University; completed requirements for Master of Science in Mechanical Engineering in May of 1987 from Kansas State University.

Professional Experience: Worked as a graduate research assistant from January 1985 to February 1987.

A B-SPLINE SURFACE MODELER WITH INTERACTIVE MODIFICATION OF THE CONTROL MESH

by

CARL ANDREW KEMNER

B.S. Kansas State University, 1984

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Mechanical Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1987

ABSTRACT

The objective of this research was to demonstrate how the B-spline approximation could be used to generate and display 3 dimensional surfaces and to develop a method to locally modify the surface representation.

The work resulted in an interactive program to generate, display, and modify, integral or rational, non-periodic B-spline surfaces through order 10. The modification module utilized the inherent 2 dimensional characteristics of the parametric surface representation to move a pointer on the 3 dimensional surface. The location of the pointer was used to specify a finite set of control points which could be used to modify the surface by changing their positions in space.

The program ran on the Kansas State University Computing Center's mainframe computer using a sub set of PLOT 10 IGL subroutines and a Selenar Hirez 100 graphics terminal as the user interface.