Chapter 7.

Curves and Surfaces

Topic: *Construction Geometry, *Model Geometry

This chapter discusses the types of curves and surfaces supported in ACIS and the classes used to implement them.

Curves

Topic: *Construction Geometry, *Model Geometry

ACIS supports these general types of curves:

Analytic curve	A simple curve that can be wholly represented by a simple algebraic formula. Simple analytic geometry can generally be analyzed up to, but not beyond, the fourth degree equation.
Interpolated curve	A curve that is the general representation of any curve that is not defined in ACIS by one of the analytic curves (i.e., by an explicit equation), but instead by reference to other geometric entities.

The curve class is the base construction geometry class for curves, and other curve construction geometry classes are derived from it. The curve class is abstract construction geometry, while its children are specific construction geometry.

The following classes are derived from the curve class:

straight . . . . Position plus direction. The magnitude is stored, but it is always 1.

The equation of a straight line, for instance, is:

$Pt = P + t*dir$

Where P is the root position, dir is a direction vector, t is a parameter for the curve space, and Pt is the 3D position of the parameter t on the line.

ellipse . . . . Represents ellipses. A circle is a special type of ellipse.

intcurve	Represents an interpolated curve. Any curve that cannot be expressed as an ellipse or a straight line can be expressed as an interpolated curve.
degenerate_curve	This special curve class is used by skinning and lofting. It is intended to provide a way to build skin or loft surfaces that come to a point at either end.
undefc	This special curve class denotes a curve that is undefined except for its end points, for which there are explicit positions, directions, and curvatures.

There are model geometry classes (with uppercase names) that correspond to these construction geometry classes. The CURVE class is the base model geometry class for curves, and other curve model geometry classes are derived from it. The CURVE class is abstract model geometry, while its children are specific model geometry.

Analytic Curve

Topic: *Construction Geometry, *Model Geometry

An *analytic curve* represents a simple curve. These curves can be represented analytically by an equation (algebraic formula). ACIS implements a certain set of analytic curves with the construction geometry classes ellipse and straight and the model geometry classes ELLIPSE and STRAIGHT.

Interpolated Curve

Topic: *Construction Geometry, *Model Geometry

An *interpolated curve* (also known as an *intcurve* because of its construction geometry class intcurve) may represent an intersection between two surfaces, the projection of a curve onto a surface, or any other general curve that is not defined by an explicit equation. An interpolated curve is used for the intersection of two surfaces where there is not a "closed-form" representation available, for exact spline curves, and for silhouette curves.

An interpolated curve, also called an object space curve, is a mapping from an interval of the real line into a 3D real vector space (object space). This mapping is continuous, and one-to-one, except possibly at the ends of the interval whose images may coincide. It must be differentiable twice, and the first derivative must be continuous and non–zero.

If the two ends of the curve are different in object space, the curve is open (refer to Figure 7-1). If they are the same, it is closed. If, in addition, the curve joins itself smoothly, the curve is periodic, with its period being the length of the interval on which it is primarily defined. A periodic curve my be evaluated outside its principal range.

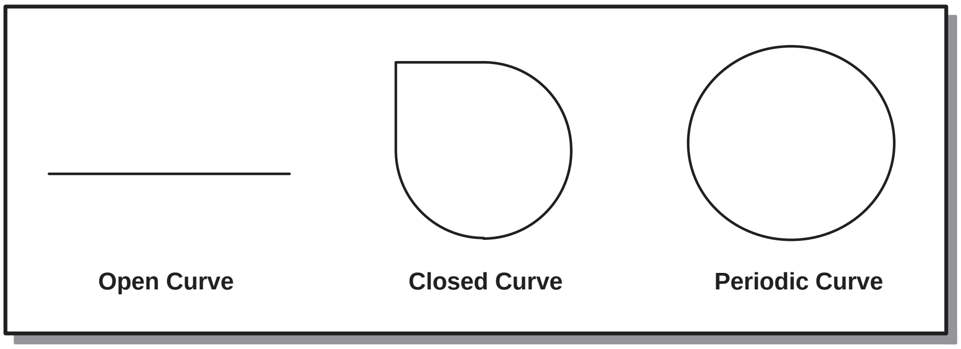

Figure 7-1. Curve Definitions

Classes

The intcurve construction geometry class provides an abstraction of the concept of a parametric representation of an interpolated curve. This interpolated curve may be either an exact curve or an approximate (3D B-spline) curve that is a fit to a true curve to within some fit tolerance.

There are two ways to define a 3D B-spline curve:

1. Directly define a curve.
2. Approximate a surface/surface intersection whose primary definition is the surfaces.

Both form part of an intcurve. The fit_tolerance (an upper limit on the distance between the true curve and the approximating curve) is zero in the first case, and positive in the second.

The intcurve class contains a reversed-sense flag and a pointer to another data structure. The flag indicates whether or not the sense of the curve is reversed with respect to the underlying spline. The pointer is to a data structure defined by the int_cur class (or some class derived from it) and contains the bulk of the information about the curve. This pointer to a separate data structure serves two purposes.

First, when duplicating an intcurve, the copy simply points to the original int_cur, thus avoiding copying the bulk of the data. A *use count* is maintained in each int_cur, which allows automatic duplication if modifying a shared int_cur, and deletion of any int_cur no longer accessible.

Second, the int_cur class contains virtual functions to perform all the operations defined for intcurve that depend on the method of definition of the true curve. New curve types are defined by declaring and implementing derived classes. The intcurve and everything using it require no changes to utilize the new definition.

The int_cur class contains the following curve defining information:

- A use count indicating the number of times the int_curve is referred.
- A pointer to a bs3_curve, which represents a spline approximation to the true curve. Refer to the section *Parameter Space Curves and Surfaces*.
- A fitting tolerance representing the precision of the spline approximation to the true curve. This is zero for an exact fit and positive for an approximation.
- Pointers to (up to) two surfaces defining the true curve. A class derived from int_cur may use these in different ways, but the true curve must lie on each of the non-NULL surfaces, if any. Either or both surface may be NULL.
- Pointers to 2D parameter space curves with respect to the defining surfaces described above. If the corresponding surface does not exists or is not parametric, this pointer is NULL.

When a surface is intersected with another surface, the intersection routines return an intcurve as part of the result. This contains the intersection approximated as a 3D parametric curve, *(x,y,z) = F(t),* for *a* <= *t* <= *b,* where *a* and *b* are the parameter values at the points on the curve corresponding to the ends of the edge. Figure 7-2 illustrates surface/surface intersection curves.

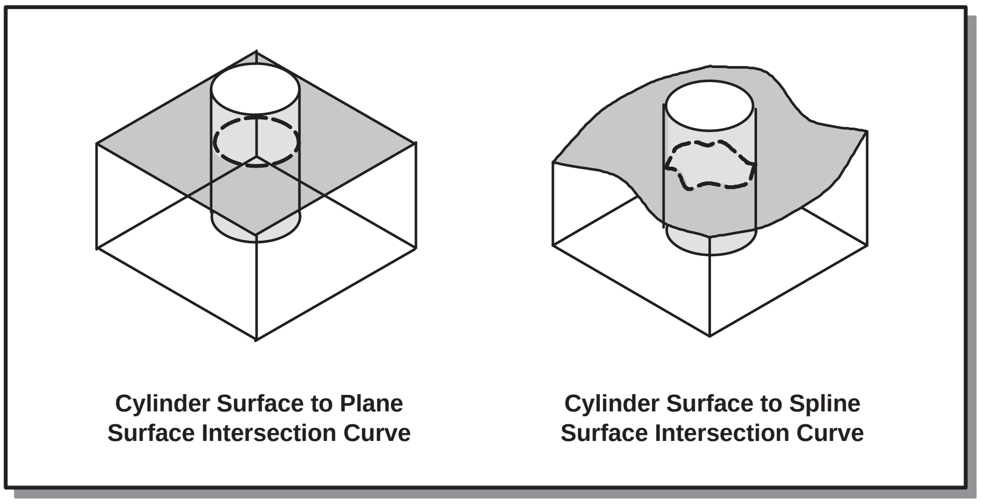
Cylinder Surface to Plane
Surface Intersection Curve
Cylinder Surface to Spline
Surface Intersection Curve

Figure 7-2. Surface to Surface Intersections

The parameter values are stored in the edge record as start_param and end_param for the start and end vertices. If the edge sense is forward along the curve, the edge stores (start_param, end_param) as (*a*, *b*). If the edge sense is reversed with respect to the curve, the edge stores (start_param, end_param) as (-*b*, -*a*). Thus, start_param is less than end_param in both cases.

If one or both of the surfaces are splines, the intcurve also contains one or two 2D parametric curves of the form, (*u*,*v*) = *G(t),* for *a* <= *t* <= *b*.

The intersection process produces the information for the 2D and 3D curve(s) at the same time, and so it is convenient to return all the curves in a single intcurve.

If the fit tolerance is not zero, the bs3_curve records an approximation to the intersection curve. When necessary, points on the approximate curve are relaxed until they lie on both surfaces (always present). The sense of the bs3_curve at a point on the curve is in the direction of the cross product N_a x N_b where N_a and N_b are the normals to the surfaces *a* and *b* at that point. The bs2_curves, if present, have the same sense as the bs3_curve.

If the fit_tolerance is zero, the bs3_curve stands alone. It is exact and no surfaces or bs2_curves are present. This form of intcurve is useful, for example in a parametric curve associated with an edge of a wire, where the parametric curve has been defined separately—perhaps by specifying the control points of its polygon.

Restrictions

ACIS does not distinguish between the curvatures on either side of a point; therefore, curves used for direct definition (zero-fit tolerance) must have a continuous curvature.

No assumption is made about the relationship between parameter values and object space distances, and parameterization is transformation-independent. However, it is advantageous for the parameterization to be as homogeneous as possible, because various iterative processes are likely to be more reliable and faster.

Surfaces

Topic: *Construction Geometry, *Model Geometry

Surfaces can be planar, cylindrical, conical, spherical, toroidal, or sculptured. Cylindrical and conical surfaces can be circular or elliptical.

ACIS supports these general types of surfaces:

Analytic surface	A simple surface that can be wholly represented by a simple algebraic formula. Simple analytic geometry can generally be analyzed up to, but not beyond, the fourth degree equation.
Spline surface	A surface that cannot be directly represented in ACIS by one of the simple analytic surfaces (i.e., by a simple analytic formula), but which can be indirectly represented by an ordered list of analytic formulae, by segmenting the curve or surface with control points.

The surface class is the base construction geometry class for surfaces, and other surface construction geometry classes are derived from it. The surface class is abstract construction geometry, while its children are specific construction geometry.

The following classes are derived from the surface class:

plane	Defines a planar surface.
cone	Defines a cylindrical or a conic surface, both of which can also be circular or elliptical.
sphere	Defines a spherical surface. It mathematically defines a sphere in both *xyz* object space and in *uv* parameter space.
torus	Defines a toroidal surface.
spline	Defines a sculpted surface. This is a surface beyond the scope of a simple analytic surface.
stripc	This special surface class defines a strip curve, which is a surface defined in a neighborhood of and passing through a given object–space curve.

There are model geometry classes (with uppercase names) that correspond to these construction geometry classes. The SURFACE class is the base model geometry class for surfaces, and other surface model geometry classes are derived from it. The SURFACE class is abstract model geometry, while its children are specific model geometry.

Analytic Surface

Topic: *Construction Geometry, *Model Geometry

An analytic surface represents a simple surface. These surfaces can be represented analytically by an equation (algebraic formula). ACIS implements a certain set of analytic surfaces with the construction geometry classes cone, plane, sphere, and torus. Other types of analytic surfaces, such as an ellipsoid, are implemented with the spline construction geometry class. There are corresponding model geometry classes (uppercase names).

Spline Surface

Topic: *Construction Geometry, *Model Geometry

In ACIS, a spline surface, also known as a parametric surface, is a mapping from a rectangle within a 2D real vector space (parameter space) into a 3D real vector space (object space), as shown in Figure 7-3. This mapping must be continuous and one-to-one, except possibly at the boundary of the rectangle in parameter space. It must be differentiable twice, and the normal direction must be continuous, though the derivatives need not be. The positive direction of the normal is in the sense of the cross product of the partial derivatives with respect to *u* and *v*, in that order. At any point on the surface, the normal points toward the part of the neighborhood that is outside the surface.

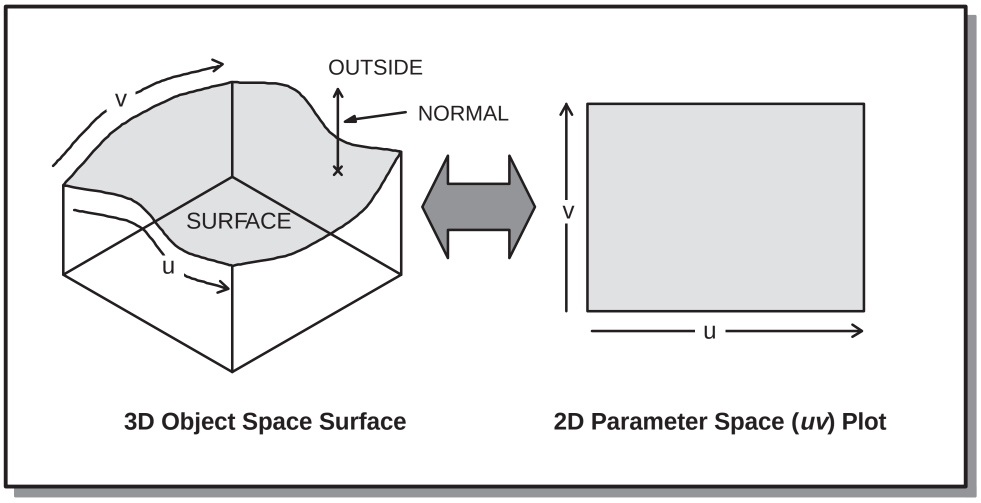

Figure 7-3. Surface Mapping

A position in parameter space is represented in ACIS by a structure called a SPApar_pos, (parameter position) that contains two real coordinates, usually referred to as *u* and *v*. A displacement in parameter space is represented as a SPApar_vec, (parameter vector). A unit direction is a SPApar_dir (parameter direction). Equivalent constructs in object space are SPAposition, SPAvector, and SPAunit_vector.

A pair of opposite sides of the rectangle may map into identical lines in object space (Figure 7-4). In this case, the surface is closed in the parameter direction normal to those boundaries. If the parameterization and derivatives also match at these boundaries, the surface is periodic in the parameter direction. The line in object space corresponding to the coincident boundaries is the seam of a periodic surface. A surface periodic in one direction but not in the other is a cylindrical surface. A surface that is periodic in both directions is a toroidal surface (Figure 7-5).

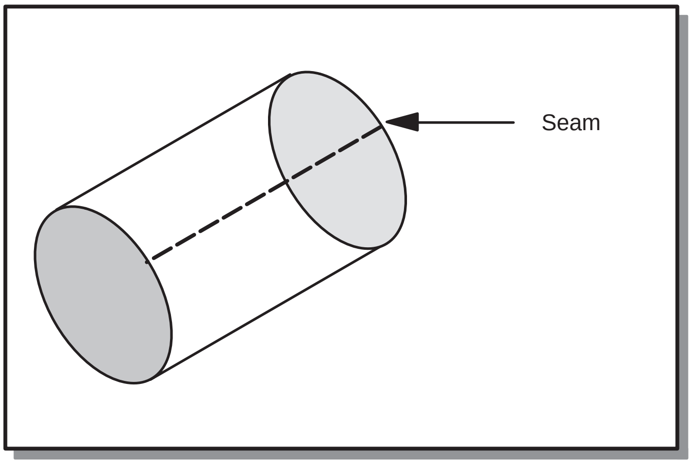

Figure 7-4. Surface Periodic in One Direction

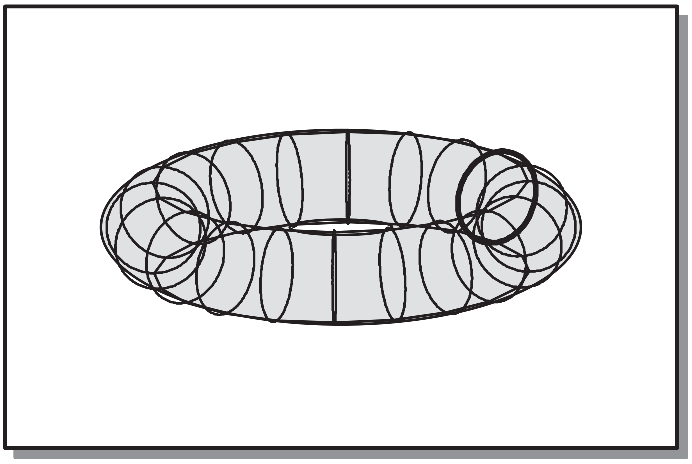

Figure 7-5. Periodic in Both Directions

If a surface is periodic in one parameter direction, it is defined for all values of that parameter. A parameter value outside the domain rectangle is brought within the rectangle by adding a multiple of the rectangle's width in that parameter direction, and the surface evaluated at that value. If the surface is periodic in both parameters, it is defined for all parameter pairs (u,v), and both parameters are reduced to standard range.

One side of the rectangle may map into a single point in object space (Figure 7-6). This point is a parametric singularity of the surface. If the surface normal is not continuous at this point, it is a surface singularity.

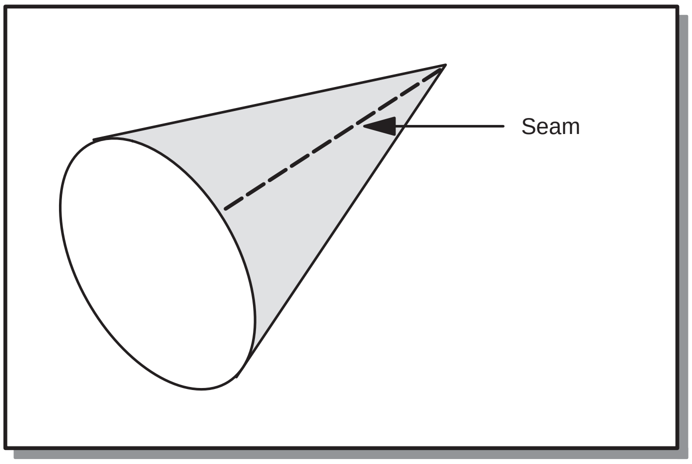

Figure 7-6. One End Mapped to a Single Point

Classes

Within ACIS, the spline construction geometry class represents a sculptured surface that contains:

- A pointer to an internal class description, called spl_sur (spline surface).
- A reversed-sense bit that, if TRUE, indicates that the sense of the spl_sur is reversed.

spl_sur contains a bs3_surface that is a pointer to a rational or nonrational nonuniform B-spline surface in the underlying sculptured surface package.

Permanent model geometry within an ACIS model is represented by the SPLINE class that contains a pointer to a spline defining the sculptured surface geometry.

Restrictions

The evaluation of mass properties in ACIS requires that the surface's first derivatives be continuous. Certain iterative processes use the derivatives for estimating the next test value, and may converge slowly, or not at all, if there are significant discontinuities in the derivatives, or if there are large and/or rapid variations in the magnitude of the derivatives. As a rule of thumb, avoid variations of more than an order of magnitude in the magnitude of first derivatives across the surface; however, such variations are tolerated if initial estimates are sufficiently accurate. Parametric singularities should be avoided, since they are likely to cause problems. Surface singularities that interfere with ACIS topological entities must not exist.

ACIS correctly handles faces that span the seam(s) of a periodic surface, but not those that form a complete band around the (distorted) cylinder or torus.

The parameterization of a surface is unaffected by transformation in object space, and so (unlike earlier versions) is not required to be a distance measure. It is still advisable that the parameterization not be extreme, and particularly that the parameterization in the two directions be of similar scale.

Parameter Space Curves and Surfaces

Topic: *Construction Geometry, *Model Geometry

To achieve functional independence for various free form curves and surfaces derived from intcurve, spline, and pcurve, ACIS provides three classes for parameter space curve and surface representations:

bs3_curve	Represents a curve derived from an intcurve. The representation is a NURBS type that handles open, closed, and periodic rational and nonrational curve representations. If a new curve is derived from the intcurve class, a method must be provided to calculate at least an approximate bs3_curve for the new curve. This means that the bs3_curve representing an ACIS defined spline curve may not be exact.
bs2_curve	Has same representation as the bs3_curve, except that it is two dimensional.
bs3_surface	Represents surfaces derived from a spline. The representation is a NURBS type that handles open, closed, and periodic rational and nonrational surface representations. If a new surface is derived from class spline, a method must be provided to calculate at least an approximate bs3_surface for the new surface. This means that the bs3_surface representing an ACIS defined spline surface may not be exact.

Use the methods defined in the base classes for all queries about the spline curves and spline surfaces. As in ACIS, the internal representation may not be an exact one. ACIS does special processing in the base class to return exact results. Also, if a new surface type or curve type is derived from the base classes, the derived class must support the methods declared virtual in the base classes.

A parameter space curve (bs2_curve) is a mapping from an interval of the real line into a 2D real vector space (parameter space). This mapping is continuous and one-to-one, except possibly at the ends of the interval whose images may coincide. It must be differentiable, and the direction of the first derivative with respect to the parameter must be continuous. This direction is the positive sense of the curve.

Parameterization of curves is transformation-independent. Therefore, the transformation of a bs2_curve is a NULL operation.

A bs2_curve is always associated with a surface that maps the parameter space image into 3D real space (object space). Thus, the two mappings together are considered to be a single mapping from a real interval into object space. Most of the properties of a parameter space curve relate to this combined mapping.

If the two ends of the curve are different in object space, the curve is open (Figure 7-1). If they are the same, it is closed. If, in addition, the curve joins itself smoothly, then the curve is periodic, with its period being the length of the interval on which it is primarily defined. A periodic curve is defined for all parameter values, by adding a multiple of the period to the parameter value so that the result is within the definition interval, and evaluating the curve at that resultant parameter. The point at the ends of the primary interval is the seam. If the surface is periodic, then a closed or periodic parameter space curve may not be closed in parameter space, but its end values may differ by the surface parameter period in one or both directions.

A bs2_curve is always associated with a bs3_curve lying in (or fitted to) the surface. It assists in the determination of the surface parameter values corresponding to object space points on the 3D curve. It does this by using the parameter value on the 3D curve to evaluate the 2D curve giving an approximation to the surface parameter values for iterative refinement. For this reason, a bs2_curve must always have the same parameter range as its associated bs3_curve, and its internal parameterization must be similar (though not necessarily identical) to that of the bs3_curve. A bs2_curve may have the same sense as its associated bs3_curve, or be opposite. In the latter case, the parameterization is negated one to the other.

The parameter curve (pcurve) is a 2D spline curve in the bi-parametric space of a parametric surface. A pcurve is attached to each coedge of a face lying on a parametric surface. It is separate from the intersection curve (intcurve) associated with the edge of the coedge, but also approximates the true intersection curve.

Thus, at an edge common to two faces, each lying on a parametric surface, there is an intersection curve (a 3D spline in model space) and two pcurves, one in each of the parametric surfaces.

The PCURVE class is derived directly from the ENTITY class and contains:

- An unsigned integer to record a use count.
- An integer type (*n*).
- Either a pcurve (if *n* is 0) or a reference to a CURVE (if *n* is not 0).

The PCURVE class records the geometry of a coedge in the parameter space of the face on which it borders. A PCURVE is attached as geometry to a coedge of the face. Each coedge of a face lying on a spline surface must reference a PCURVE.

The parameterization of the PCURVE is similar to that of the 3D curve. Both have the same limits (or are negated if they are in opposite directions). One traces out an edge of a face lying on a parametric surface by setting values of t in $F(t)$ or in $S(G(t))$, where a point on the parametric surface is given by $(x,y,z) = S(u,v)$.

In either case, a curve is obtained that approximates the true curve of intersection. The two curves are similar but not identical. In general, they differ by small amounts in position and in parameterization.

For a PCURVE stored in this way, the integer n in the PCURVE is set to +1, –1, +2 or –2, indicating that the 2D parametric curve is found within the referenced INTCURVE. If n is negative, the bs2_curve within the intcurve is negated.

For a PCURVE with $n = 0$, the details are held in a pcurve that contains:

- A pointer to an internal description termed a par_cur.
- A sense flag (logical) that, if TRUE, indicates that the par_cur is reversed.

The par_cur contains:

- A 2D bs2_curve $(u,v) = G(t)$.
- A fit_tolerance.
- A pointer to the spline surface with respect to the defined 2D curve surface.

This form of PCURVE is employed for example when a parametric surface is constructed by sweeping a parametric curve along a curve. The PCURVE is a constant parameter line along the edge of the swept surface patch.

Curve Sense

Topic: *Construction Geometry, *Model Geometry

Care must be used to ensure that the sense of the parameterization of curves and parameter space curves is correct. The parameter limits of the curve of an edge and the parameter space curves of any of its coedges are found from the edge as its start and end parameters that correspond to the start and end vertices of the edge.

Recall a pcurve is attached to each coedge of a face lying on a parametric surface. An edge common to two faces, each lying on a parametric surface, has an intersection curve (a 3D spline in model space) and two pcurves, one in each of the parametric surfaces.

Consider the edge first. If the edge has the opposite sense to its underlying curve, the edge parameter interval (a,b) recorded in the edge is taken to be the interval $(-b,-a)$ when finding points along the curve. The stored form of the curve must be consistent with this convention.

In the case of the model geometry class INTCURVE, there is a further sense logical within the stored intcurve so that the intcurve may be reversed without altering the details of the stored bs3_curve, and the parameter values are negated and interchanged if so.

Now, trace out the points along a coedge, that is in the sense of the coedge as it bounds a face (metal on the left). It is possible to trace out points along the edge and then take the set of points in reverse order if the coedge sense bit records the coedge as reversed with respect to the edge.

Alternatively, if the coedge bounds a face lying on a spline surface, it is possible to find the same set of points (strictly, a similar set of points) from the coedge and its parameter space curve. The procedure is to find the parameter limits (*a*,*b*) from the edge, reverse them to (-*b*,-*a*) if the coedge is reversed with respect to the edge, and then present these limits to the stored pcurve to obtain the points ordered along it in the desired sense.

As with an intcurve, the stored form of a private pcurve contains a sense bit so that a private pcurve is reversed without having to change the underlying details of the stored 2D parametric curve or 2D parametric surface. As described above, for pcurves associated with an existing model geometry class INTCURVE, sense reversal is denoted by a negative value of *n*, where *n* refers to the *n*th pcurve of the referenced intcurve.

Because the interpretation of the curve stored is governed by all of the intervening sense bits, it is is essential to understand these details when setting bs2_curves into pcurves or bs3_curves into intcurves.

Continuity Requirements

Topic: *Mathematics, *Construction Geometry

ACIS prefers that curves and surfaces generally be G2 continuous, but will allow G1 continuity at the knots of curves and surfaces. The C++ class discontinuity_info records discontinuity information. It is used internally by ACIS in the curve and surface extension and intersection algorithms.

Some modeling operations (such as offsetting, shelling, and blending) may fail on surfaces that are not G1 continuous. The evaluation of mass properties in ACIS requires that curves be C1 continuous.

Refer to Chapter 3, *Math Foundation*, for definitions of the types and degrees of continuity.

Extending Curves and Surfaces

Topic: *Construction Geometry, *Model Geometry

Many functions within ACIS must make curves and surfaces larger. For example, the blending end capping algorithm and the Local Operations Component must extend curves and surfaces to successfully complete their operations.

Generally this is done by extending the natural definition of the entity and extending any supporting curves or surfaces as necessary. For example, the extension of an offset intersection curve defining the spine of a blend is done by extending the two surfaces that define the curve and intersecting them. Extending exact interpolated curves and exact spline surfaces is done by adding a tangential straight line or a tangential ruled surface, respectively. Extended exact interpolated curves and spline surfaces provide examples of geometry which is only C1 continuous.

Subsetting Curves and Surfaces

Topic: *Construction Geometry, *Model Geometry

Sometimes it is beneficial to make a curve or surface smaller without losing the underlying definition. Intersection algorithms can be more efficient and faster if they limit their search to a subset of the underlying geometry.

Curve and surface subsetting provides a solution. Class methods are provided to construct a subsetted curve or surface, change its subset, or revert to the original underlying curve or surface definition.

Subsetting Option

The option subsetting allows specification of the level of subsetting when trimming. This enables the use of the same underlying spl_sur for different faces. This provides a minor performance enhancement and a reduction in the size of SAT files.

Isoparameter Curves

If a surface is subsetted, it does not span the entire range of the underlying, defining surface. Isoparameter curves on such a subset surface should not span the entire defining surface either. The methods to generate isoparameter curves on sum_spl_surs take this reduced range into consideration.

Face Merging

When subsetted surfaces are merged, the new subset range (of the surface underlying the newly created, merged face) is the union of the two previous subset ranges. When determining if an edge lies on a seam of a subsetted, closed surface, the determination is made as if the surface were not subsetted.

Parameter Values on Subset Intcurves

If an intcurve is closed, there are two valid parameter values for a position on the seam of the curve. If the curve is subsetted such that one of the ends is not in the subset range, then there is only one valid parameter value. The method intcurve::param returns the corrected parameter value when the intcurve has been subsetted in this manner.