Chapter 19.

Functions bs3_curve Ma thru Rz

Topic: Ignore

bs3_curve_make_cur

Function: Spline Interface, Construction Geometry

Action: Creates a curve from a portion of a general ACIS curve.

Prototype:

```
bs3_curve bs3_curve_make_cur (
    curve const& cur,        // given curve
    double start_param,      // start parameter
    double end_param,        // end parameter
    double requested_tol     // required fit tolerance
        = 0,
    double& actual_tol       // actual tolerance used
        =*(double*)NULL_REF
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "kernel/kerngeom/curve/curdef.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/bs3_crv/sp3crtn.hxx"
```

Description: The parameter range of the new curve will be from the given start parameter to the given end parameter. It may not match the original parameterization in between.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3_crv/sp3crtn.hxx

Effect: Changes model

bs3_curve_make_ell

Function: Spline Interface, Construction Geometry

Action: Creates a spline curve from an elliptical arc.

Prototype:

```
bs3_curve bs3_curve_make_ell (
    ellipse const& ell,      // given ellipse
    double start_param,      // start parameter
    double end_param,        // end parameter
    double                   // requested tolerance
         = 0,
    double& actual_tol       // returned actual
        =*(double*)NULL_REF  // tolerance used
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "kernel/kerngeom/curve/elldef.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/bs3_crv/sp3crtn.hxx"
```

Description: The parameter range of the new curve will be from the given start parameter to the given end parameter. The parameterization of the curve is not defined, but will be as uniform as is practicable.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3_crv/sp3crtn.hxx

Effect: Changes model

bs3_curve_make_int

Function: Spline Interface, Construction Geometry

Action: Creates a spline curve from an intcurve.

Prototype:

```
bs3_curve bs3_curve_make_int (
    intcurve const& cur,     // given curve
    double start_param,      // start parameter
    double end_param,        // end parameter
    double requested_tol     // requested fit
        = 0,                 // tolerance
    double& actual_tol       // returned actual
        =*(double*)NULL_REF  // tolerance used
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "kernel/kerngeom/curve/intdef.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/bs3_crv/sp3crtn.hxx"
```

Description: This routine is not normally very useful, but called by bs3_curve_make_cur.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3_crv/sp3crtn.hxx

Effect: Changes model

bs3_curve_make_rho_conic

Function: Spline Interface, Construction Geometry

Action: Creates a general conic curve segment given three distinct points and a *rho* value.

Prototype:

```
bs3_curve bs3_curve_make_rho_conic (
    SPAposition const& start,   // start point
    SPAposition const& tan_int, // point where
tangents intersect
    SPAposition const& end,     // end point
    double rho                  // controls fullness,
        = 0.5,
    double                      // requested
         = 0,                   // tolerance
    double& actual_tol          // returned actual
        =*(double*)NULL_REF     // tolerance used
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "baseutil/vector/position.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/bs3_crv/sp3crtn.hxx"
```

Description: This segment runs from the start point to the end point, and is tangent to the line from the start to the intersection point at the start, and to the line from the intersection point to the end.

The “fullness” of the curve is controlled by the rho value. The line segment is constructed from the mid point of the chord between start and end points to the tangent intersection point. The “shoulder point” is the point that divides that line segment in the ratio rho:(1 – rho). The required conic is the one obeying the end conditions that passes through the shoulder point.

rho must be such that 0 < rho < 1. If rho is 0.5 (default), the conic is parabolic. If rho is less than 0.5, the conic is a segment of an ellipse, becoming more and more flattened as rho tends towards 0.0. If rho is greater than 0.5, the conic is a segment of one branch of a hyperbola, with an increasingly sharp bend near the tangent intersection point as rho nears 1.0.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3_crv/sp3crtn.hxx

Effect: Changes model

bs3_curve_make_str

Function: Spline Interface, Construction Geometry

Action: Creates a parametric curve that is geometrically the same as the portion of the given straight line between the given parameter values.

Prototype:

```
bs3_curve bs3_curve_make_str (
    straight const& str,     // given straight line
    double start_param,      // start parameter
    double end_param,        // end parameter
    double                   // requested tolerance
         = 0,
    double& actual_tol       // returned actual
        =*(double*)NULL_REF  // tolerance used
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "kernel/kerngeom/curve/strdef.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/bs3_crv/sp3crtn.hxx"
```

Description: The parameter range of the new curve must be from the given start parameter to the given end parameter. It does not have to be linear in object space, but it would be most surprising if it were not.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3_crv/sp3crtn.hxx

Effect: Changes model

bs3_curve_max_dist_to_point_bound

Function: Spline Interface, Construction Geometry

Action: Gets an upper limit to the maximum distance from a point in space to the given curve.

Prototype:
```
double bs3_curve_max_dist_to_point_bound (
    bs3_curve bs3_curve,     // given curve
    const SPAposition& P     // position
    );
```

Includes:
```
#include "kernel/acis.hxx"
#include "baseutil/vector/position.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/d3_bs3/spd3rtn.hxx"
```

Description: This routine finds the maximum distance from the point to each of the knots on the curve.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/d3_bs3/spd3rtn.hxx

Effect: Read–only

bs3_curve_mergeable

Function: Spline Interface, Construction Geometry

Action: Sets two curves so they can be merged.

Prototype:
```
void bs3_curve_mergeable (
    bs3_curve b1,            // first given curve
    bs3_curve b2             // second given curve
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/sg_bs3c/sps3crtn.hxx"
```

Description: This function reparameterizes the curves if they are rational, but doesn't match tangent magnitudes for C1 continuity. Refer to bs3_curve_connect for additional information.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/sg_bs3c/sps3crtn.hxx

Effect: Changes model

bs3_curve_mid

Function: Spline Interface, Construction Geometry

Action: Gets the mid–point of the given spline curve.

Prototype:

```
SPAposition bs3_curve_mid (
    bs3_curve bs            // given curve
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "baseutil/vector/position.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/sg_bs3c/sps3crtn.hxx"
```

Description: Refer to Action.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/sg_bs3c/sps3crtn.hxx

Effect: Read–only

bs3_curve_mid_tangent

Function: Spline Interface, Construction Geometry

Action: Gets the normalized mid–tangent to the given spline.

Prototype:

```
SPAunit_vector bs3_curve_mid_tangent (
    bs3_curve bs            // given curve
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "baseutil/vector/unitvec.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/sg_bs3c/sps3crtn.hxx"
```

Description: Refer to Action.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/sg_bs3c/sps3crtn.hxx

Effect: Read–only

bs3_curve_modify

Function: Spline Interface, Construction Geometry

Action: Modifies the curve, such that the curve passes through the given point at the given parameter value and has the given tangent direction.

Prototype:

```
logical bs3_curve_modify (
    bs3_curve bs_cur,       // given curve
    SPAposition const& pos, // position to be
                            // interpolated
    SPAunit_vector const& dir,// direction to be
                            // interpolated
    double t                // parameter value to be
                            // interpolated
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "baseutil/logical.h"
#include "baseutil/vector/position.hxx"
#include "baseutil/vector/unitvec.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/sg_bs3c/sps3crtn.hxx"
```

Description: The given curve is modified in place. Returns TRUE if the operation is successful.

It may assume that the ends to be joined match in position and direction, and that the parameterization is continuous. ACIS uses this routine only to rejoin parts of a periodic curve that have been split by bs3_curve_split, in the opposite order, so that the resultant curve starts and ends at the split point. A curve made in this way may be marked as closed by bs3_curve_join, but should not be marked periodic.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/sg_bs3c/sps3crtn.hxx

Effect: Changes model

bs3_curve_mult

Function: Spline Interface, Construction Geometry

Action: Determines if the current curve has multiple end knots.

Prototype:

```
logical bs3_curve_mult (
    bs3_curve input         // given curve
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "baseutil/logical.h"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/sg_bs3c/sps3crtn.hxx"
```

Description: This routine returns TRUE is the curve has fully multiple end knots at the beginning and end of the curve; returns FALSE otherwise.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/sg_bs3c/sps3crtn.hxx

Effect: Read–only

bs3_curve_nderiv

Function: Spline Interface, Construction Geometry

Action: Evaluates the *n*th derivative of a given bs3_curve at a given parameter value.

Prototype:

```
SPAvector bs3_curve_nderiv (
    double param,           // parameter at which
                            // derivative is desired
    bs3_curve cur,          // given curve
    int nder                // derivative number
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "baseutil/vector/vector.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/bs3_crv/sp3crtn.hxx"
```

Description: The derivative is returned. This routine is mostly used for computing the third derivative of a bs3_curve or up.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3_crv/sp3crtn.hxx

Effect: Read–only

bs3_curve_nspans

Function: Spline Interface, Construction Geometry

Action: On the assumption that the curve consists of a sequence of simple spans, gets the number of such spans.

Prototype:

```
int bs3_curve_nspans (
    bs3_curve cur           // given curve
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/bs3_crv/sp3crtn.hxx"
```

Description: The precise definition of simple is moot. But, for a piecewise polynomial curve the simple spans should be the polynomial pieces.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3_crv/sp3crtn.hxx

Effect: Read–only

bs3_curve_num_ctlpts

Function: Spline Interface, Construction Geometry

Action: Gets the number of control points in a bs3_curve.

Prototype:

```
int bs3_curve_num_ctlpts (
    bs3_curve bs            // bs3_curve
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/sg_bs3c/sps3crtn.hxx"
```

Description: Refer to Action.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/sg_bs3c/sps3crtn.hxx

Effect: Read–only

bs3_curve_offset_by_dist

Function: Spline Interface, Construction Geometry

Action: Offsets a planar bs3_curve by a specified distance.

Prototype:

```
bs3_curve bs3_curve_offset_by_dist (
    const bs3_curve bs,     // given curve
    double dist,            // distance to offset
    logical to_right,       // direction to offset
                            // (left or right)
    const SPAunit_vector&   // direction of
        plane_nor           // plane normal
        =*(SPAunit_vector*)NULL_REF
    );
```

```
bs3_curve bs3_curve_offset_by_dist (
    const bs3_curve bs,      // given curve
    double dist,             // distance to offset
    logical to_right,        // direction to offset
                             // (left or right)
    logical make_seg_list,   // curve normal
    bs3_offset_seg*&         // list of
        seg_list,            // segments
    SPAunit_vector& plane_nor// plane normal
        =*(SPAunit_vector*)NULL_REF
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "baseutil/logical.h"
#include "baseutil/vector/unitvec.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/sg_bs3c/bs3cseg.hxx"
#include "kernel/spline/sg_bs3c/sps3crtn.hxx"
```

Description: The curve is offset in the plane in which it lies, and either to the right (or left) as viewed along the curve. If an error occurs, a NULL curve is returned.

The plane_nor argument supplies the plane normal, and is used only for linear b–spline cases. In all other cases plane_nor is ignored.

Errors: If the spline is linear and plane_nor is not specified, the function returns a NULL b–spline (unsuccessful).

If the given B–spline is a degenerate B–spline the function returns a NULL B–spline (unsuccessful).

Limitations: None

Library: kernel

Filename: kern/kernel/spline/sg_bs3c/sps3crtn.hxx

Effect: Changes model

bs3_curve_offset_by_vec

Function: Spline Interface, Construction Geometry

Action: Offsets a planar bs3_curve by a vector.

Prototype:

```
bs3_curve bs3_curve_offset_by_vec (
    const bs3_curve bs,      // given curve
    const SPAvector& offset  // vector to offset by
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "baseutil/vector/vector.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/sg_bs3c/sps3crtn.hxx"
```

Description: The vector does not have to lie in the plane of the curve. If an error occurs, a NULL curve is returned.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/sg_bs3c/sps3crtn.hxx

Effect: Changes model

bs3_curve_open

Function: Spline Interface, Construction Geometry

Action: Determines if a bs3_curve is open.

Prototype:

```
logical bs3_curve_open (
    bs3_curve cur           // given curve
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "baseutil/logical.h"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/bs3_crv/sp3crtn.hxx"
```

Description: A curve is open if its end points do not meet at the same point to form a closed loop.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3_crv/sp3crtn.hxx

Effect: Read–only

bs3_curve_param_length

Function: Spline Interface, Construction Geometry

Action: Gets the signed arc length of the curve between two parameter values.

Prototype:

```
double bs3_curve_param_length (
    bs3_curve cur,          // given curve
    double start,           // parameter of start
                            // point
    double end              // parameter of end point
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/bs3_crv/sp3crtn.hxx"
```

Description: The length will be positive if the second parameter is greater than the first, and negative if it is less, provided both parameters are valid for the curve.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3_crv/sp3crtn.hxx

Effect: Read–only

bs3_curve_period

Function: Spline Interface, Construction Geometry

Action: Gets the length of the primary interval (i.e., the parametric period) that defines a periodic curve.

Prototype:

```
double bs3_curve_period (
    bs3_curve cur           // given curve
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/bs3_crv/sp3crtn.hxx"
```

Description: Returns exactly 0 for any nonperiodic curve.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3_crv/sp3crtn.hxx

Effect: Read–only

bs3_curve_periodic

Function: Spline Interface, Construction Geometry

Action: Determines if the curve is smoothly closed and is an endless loop.

Prototype:
```
logical bs3_curve_periodic (
    bs3_curve cur           // given curve
    );
```

Includes:
```
#include "kernel/acis.hxx"
#include "baseutil/logical.h"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/bs3_crv/sp3crtn.hxx"
```

Description: If the curve is periodic, it is defined for all parameter values.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3_crv/sp3crtn.hxx

Effect: Read–only

bs3_curve_perp

Function: Spline Interface, Construction Geometry

Action: Gets the position, tangent direction, and parameter value at the foot of a perpendicular dropped from a point to the spline curve.

Prototype:

```
void bs3_curve_perp (
    SPAposition const& pos,      // given point
    bs3_curve cur,               // given curve
    SPAposition& foot,           // returned foot of
                                 // perpendicular
    SPAunit_vector& tan,         // returned curve
                                 // tangent
    SPAparameter const&          // guess value
        param_guess              // for parameter
        =*(SPAparameter*)NULL_REF,
    SPAparameter& param_actual   // returned actual
        =*(SPAparameter*)NULL_REF,// parameter used
    double quick_exit_dist_tol   // if non-zero, used
        = 0.0                    // to stop iteration
                                 // when foot is
                                 // closer than tol to
                                 // pos
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "baseutil/vector/param.hxx"
#include "baseutil/vector/position.hxx"
#include "baseutil/vector/unitvec.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/bs3_crv/sp3crtn.hxx"
```

Description:

All return parameters are optional (i.e., they may be NULL references, meaning that the value is not required). If an approximate parameter is supplied, the returned point will be the one near (in some sense) to the specified parameter. If not, it will be the point nearest in object space to the given one.

For an open curve, there may be no perpendicular at all. In this case, one end of the curve is returned and the tangent and parameter at that point are returned as usual.

If a non–zero quick_exit_dist_tol is supplied, this value is used to stop the point perp iteration as soon as the near pt and pos are within quick_exit_dist_tol of each other.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3_crv/sp3crtn.hxx

Effect: Read–only

bs3_curve_planar

Function: Spline Interface, Construction Geometry

Action: Determines whether the curve is planar, within the specified tolerance.

Prototype:

```
int bs3_curve_planar (
    const bs3_curve& bc,     // given curve
    SPAunit_vector& normal,  // normal to plane
    double tolerance         // within this tolerance
        = SPAresabs
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "baseutil/vector/unitvec.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/sg_bs3c/sps3crtn.hxx"
```

Description: The routine returns one of the following values:

–2 = Curve is a point or a degenerate spline.
–1 = Curve is a straight line, normal not set.
0 = Curve is nonplanar.
1 = Curve is planar, and normal is set.

If the given curve is planar, it sets the argument normal to be the normal of the plane in which the given spline lies.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/sg_bs3c/sps3crtn.hxx

Effect: Read–only

bs3_curve_position

Function: Spline Interface, Construction Geometry

Action: Evaluates the curve at the given parameter value.

Prototype:

```
SPAposition bs3_curve_position (
    double param,           // given parameter value
    bs3_curve cur           // given curve
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "baseutil/vector/position.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/bs3_crv/sp3crtn.hxx"
```

Description: Normally, this is implemented as a call to bs3_curve_eval.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3_crv/sp3crtn.hxx

Effect: Read–only

bs3_curve_rail_from_cen

Function: Spline Interface, Construction Geometry

Action: Gets a rail curve to sweep for surface construction.

Prototype:

```
bs3_curve bs3_curve_rail_from_cen (
    bs3_curve input,        // given curve
    SPAunit_vector up_dir,  // up direction, normal
                            // to the spline at start
    double twist_ang        // twist angle
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "baseutil/vector/unitvec.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/sg_bs3c/sps3crtn.hxx"
```

Description: To establish the twist angle, this routine uses the angle of rotation (in radians). The routine can also be used to get the lateral curves for sweep.

As per the definition of the sweep, the up_dir is propagated along the center curve. up_dir is the normal to the plane formed by the center a nonplanar B–spline. Any normal is acceptable.

The rail is constructed so that it forms a Frenet frame as it goes along the path curve to make up a sweep surface.

For this routine, the center curve must be a cubic B–spline.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/sg_bs3c/sps3crtn.hxx

Effect: Read–only

bs3_curve_range

Function: Spline Interface, Construction Geometry

Action: Gets the primary interval on which the curve is defined, which in open and nonperiodic closed curves is the actual domain.

Prototype:

```
SPAinterval bs3_curve_range (
    bs3_curve cur           // given curve
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "baseutil/vector/interval.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/bs3_crv/sp3crtn.hxx"
```

Description: Refer to Action.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3_crv/sp3crtn.hxx

Effect: Read–only

bs3_curve_rational

Function: Spline Interface, Construction Geometry

Action: Determines whether a spline curve is rational.

Prototype:

```
logical bs3_curve_rational (
    bs3_curve input         // given curve
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "baseutil/logical.h"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/sg_bs3c/sps3crtn.hxx"
```

Description: This routine returns TRUE if it is rational; otherwise returns FALSE.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/sg_bs3c/sps3crtn.hxx

Effect: Read–only

bs3_curve_rem_extra_knots

Function: Spline Interface, Construction Geometry

Action: Removes redundant knots from a bs3_curve.

Prototype:

```
void bs3_curve_rem_extra_knots (
    bs3_curve bs3,          // curve
    double tol              // tolerance
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/sg_bs3c/sps3crtn.hxx"
```

Description: This only removes knots that are truly redundant, i.e. when their removal will not affect the shape or parameterization of the curve by more than the given tolerance.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/sg_bs3c/sps3crtn.hxx

Effect: Changes model

bs3_curve_reparam

Function: Spline Interface, Construction Geometry

Action: Reparameterizes the given curve in place.

Prototype:

```
void bs3_curve_reparam (
    double start,           // start parameter
                            // desired
    double end,             // end parameter desired
    bs3_curve cur           // given curve
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/bs3_crv/sp3crtn.hxx"
```

Description: Reparameterizes the given curve in place by a linear transformation so that its primary interval of definition is from the start to the end parameters given (which must be in increasing order). Each new parameter value will be the appropriate linear function of the old parameter value.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3_crv/sp3crtn.hxx

Effect: Changes model

bs3_curve_restore

Function: Spline Interface, Construction Geometry, SAT Save and Restore

Action: Restores a curve from a file.

Prototype:

```
bs3_curve bs3_curve_restore ();
```

Includes:

```
#include "kernel/acis.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/bs3_crv/sp3crtn.hxx"
```

Description: Reads back a representation of a parametric curve written by bs3_curve_save and constructs a duplicate of the original curve. Reading uses routines read_int, read_long, read_real, and read_string defined in kernutil/fileio/fileio.hxx.

The overloaded >> operator acts like bs3_curve_restore, but reads from a C++ style stream using stream operators, and sets the result into the second argument.

bs_2_3_spline_restore	Restore spline

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3_crv/sp3crtn.hxx

Effect: System routine

bs3_curve_reverse

Function: Spline Interface, Construction Geometry

Action: Reverses the direction of the given curve, and negates the parameterization.

Prototype:

```
void bs3_curve_reverse (
    bs3_curve cur            // given curve
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/bs3_crv/sp3crtn.hxx"
```

Description: Reverses the direction of the given curve, and negate the parameterization (so that the new primary definition interval is $[-b, -a]$ if the original was $[a, b]$).

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3_crv/sp3crtn.hxx

Effect: Changes model