Report No. 87 01 14 - 4

NTNF-VF 24.9660

REFERENCE MANUAL

APS - B-SPLINE LIBRARY

VERSION 4.0

by Vibeke Skytt and Tor Dokken

1987-11-10

SENTER FOR INDUSTRIFORSKNING

SENTER FOR INDUSTRIFORSKNING
Forskningsveien 1, Postboks 124 Blindern, 0314 Oslo 3. Tlf. (02) 45 20 10

REPORT No : 87 11 10 - 7

This report : CONFIDENTIAL
This page : OPEN

REFERENCE MANUAL
APS - B-SPLINE LIBRARY VERSION 4.0

by Vibeke Skytt and Tor Dokken

APS Sculptured Surfaces is a FORTRAN subroutine package for definition and manipulation of sculptured surface geometry. This report contains the description of the top level subroutines of the B-spline library of the subroutine package version 4.0.

PROJECT : APS Geometrisk Modellering

CLIENT : NTNF/VF

CLIENT'S REF : VF 24.9660

Department : CAD/CAM and Robotics
R&D field : Geometric Modelling

SI's project manager : Tor Dokken
Project supervisor : Stig Ulfsby

Approved Date : 06.11.87 Stig Ulfsby (sign)

---

4 index words:

Norwegian	English
DAK/DAP	CAD/CAM
Skulpturerte flater	Sculptured Surfaces
Geometrisk Modellering	Modelling
B-splines	B-splines

Number of pages: 211

APS
advanced production system

REPORT NO: 7 APS/SCULPTURED SURFACES	DATE: 1987-11-10	NO. OF PAGES: 211

APS SUBPROJECT:

GEOMETRIC MODELLING TOOLS

SUBPROJECT ACTIVITY:

SCULPTURED SURFACE MODELLER

REPORT TITLE:

APS - B - SPLINE LIBRARY

Version 4.0

PROJECT TEAM:

L.Arneberg
T.Dokken
M.Dæhlen
P.Evensen
J.Kåsa
* V.Skytt
A.M.Ytrehus

* Author of this report

PREFACE:

Advanced production System (APS) is a joint German Norwegian CAD/CAM research project.

The project is sponsered by:

- * Royal Norwegian Council for Scientific and Industrial Research
  Norway
- * Bundesministerium für Forschung und Technik (BFMT)
  BRD
- * Deminex
  Norway / BRD

The participating research institutes are:

IPK - Fraunhofer-Institute für Produktionsanlagen
und Konstruktionstechnik
Berlin, BRD

WZL - Laboratorium für Werkzeugmaschinen und
Betriebslehre der RWTH Aachen
Aachen, BRD

SI - Center for Industrial Research
Oslo, Norway

NTH/SINTEF - Trondheim, Norway.

This report is a part of the documentation of the APS-Integrated Geometric Modeller. This has been built in coopertion between IPK and SI. The COMPAC Volume Modeller from IPK has been integrated with the APS-Sculptured Surface Modeller from SI (APS-SS).

The B-spline library of APS-SS version 4.0. is documented in this report.

Oslo 5 October 1987

Tor Dokken

Tor Dokken

M A I N   C H A P T E R S

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

1 Introduction

This report describes the user routines at the geometry calculation level of the APS sculptured surface module. These subroutines are used by the sculptured surface module of the APS Integrated Geometric Modeller. They can also be used directly, thus avoiding the data structure handling of the sculptured surface modeller. In the latter case the subroutines/program using these subroutines must take care of the data structure handling.

The APS-Integrated Geometric Modeller has been developed in a cooperation project between IPK (Berlin, BRD), Norsk Data (Mulheim, BRD and Oslo,Kongsberg Norway) and SI (Oslo, Norway) financed by the German-Norwegian Cooperation Project APS-Advanced Production System. The volume modeller COMPAC from IPK and the Sculptured Surface modeller from SI have been integrated. The data structures and application interface of the Sculptured Surface module of this integration is documented in:

SI-Report 87 01 14 - 3: Reference Manual: APS-Sculptured Surfaces.

This report documents the mathematical routines available in the APS-Sculptured Surface Module, the B-spline library. The routines in this library are self contained, they do not call routines from other packages.

The first chapter is this introduction. In chapter two we give some general comments on the philosophy of the geometry calculation subroutines and how the routines should be used. First the general B-spline format is explained, then the use of scratch arrays. The naming conventions and standard parameter names are listed as well as the standard error messages.

Chapter 3 contains different ways of defining curve geometry. Chapter 4 goes into how to calculate points and derivatives on the curves, while chapter 5 goes into the intersection problems and chapter 6 contains some additional useful subroutines.

Chapter 7 tells how to define B-spline surfaces, while chapter 8, 9 and 10 describes respectively, calculation of points and curves on surfaces, intersection routines and other B-spline routines.

Below the subroutines described, many B-spline service routines exist, these take care of the B-spline specific problems and are hidden for the users. We want to make the users think geometry and try to avoid to force the user into thinking of mathematical problems not relevant to the geometry problems he/she wants to solve.

This is version 4.0 of the B-spline library. The prior versions are documented in Report No:

87 01 14 - 1 from SI: Reference Manual: APS B-spline Library, version 3.3.

TABLE OF CONTENTS

2 General comments.

To make a general subroutine package for calculations on curves and surfaces, the following philosophy has been followed:

- Curves and surfaces can be defined in a set of different ways.

  The evaluated curves and surfaces are represented in a general B-spline format.

- All calculations performed on the curves and the surfaces assume that the curve geometry is expressed in the general B-spline format.

Parameter names and sequences are standardized. The error status returned from subroutines are standardized, as well as names on parameters. Naming conventions exist for variable names.

2.1 The format of evaluated curves and surfaces.

Parametric B-splines are used as the storage format for evaluated curve geometry due to the versatility in shapes it can represent, the compact storage format, the explicit representation of the continuity between adjacent polynomial segments and the stability of the algorithms working on the B-spline format.

2.1.1 The B-spline format for curves.

Piecewise polynomial curves can be represented in a variety of ways. One of these is the B-spline representation, which is used here in the APS sculptured surface modul. This representation consists of five different components.

t(j) - The knots of the B-spline representation. j=1,...,N+k.

C(i,j) - The coefficients/vertices of the B-spline representation. i=1,...,DIM, j=1,...,N

N - The number of vertices.

k - The polynomial order (degree+1) of the B-spline basis.

DIM - The dimension of the space in which the curve lies.

The knots of the B spline curve are a little tricky to understand. They describe the following properties of the curve:

- The parametrization of the B-spline curve.

- The continuity at the joints between the adjacent polynomial segments on the B-spline curve.

The information of the knot vector can be read in the following way:

- The k th entry in the knot vector is the parameter value of the start point of the curve.

- The (N+1)-th entry in the knot vector is the parameter value of the end point of the curve.

- The entries in the knot vector with number k+1,...,N are parameter values where one polynomial segment ends and another polynomial segment starts. The knots are always listed in increasing order. More than one knot can have the same value. If some of the internal knots have the same value as the start or the end parameter value of the curve, some of the description of the B-spline curve can be degenerate. Degenerate curve description does not affect the geometry algorithms.

- The continuity of the curve at the knot values depends on the number of knots having the same value. The number of knots having the same value is called the multiplicity, "m", of the knot value The order of continuity at a knot value is "k-m-1". ("k" is as earlier defined as the polynomial order of the B-spline curve.) If k knot values are equal, the curve is discontinuous at the specified parameter value. If the multiplicity "m"=1, the order of continuity at the knot value is "k-2".

To illustrate the properties of the description of the B-spline curves, we will look at an example.

<u>EXAMPLE.</u>

Let a B-spline curve have the following description.

Knot vector:
    1, 1, 1, 1, 2, 2.5, 2.5, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5

Vertices:
    (1,1), (2,2), (3,1), (4,2), (5,1), (6,2), (7,1), (8,2), (9,1),
    (10,2), (11,1), (12,2), (13,1), (14,2)

Number of vertices:
    N=14

Polynomial order:
    k=4

Number of knots:
    N+k = 14 + 4 = 18

Dimension of space:
    DIM=2

In Figure 2.1 the vertices and the resulting curve are shown. It is clear that not only the vertices affect the shape of the curve but also the knot values and multiplicities. Figure 2.2 shows how the knot values are mapped onto the curve and how the multiplicity of the knots affects the continuity of the curve.

Analysis of the knot vector:

- The start parameter value is 1, because the k-th entry, i.e. the 4-th entry in the knot vector is 1.

- The end parameter value is 5, because the N+1-th entry, i.e. the 15-th entry in the knot vector is 5.

  Internal joints between polynomial segments at the parameter values 2, 2.5, 3 and 4.

- Continuity up to and including second derivative at parameter value 2, because the multiplicity of the parameter value 2 is m=1, and k-m-1=4-1-1=2.

- Continuity up to and including first derivative at the parameter value 2.5, because the multiplicity of the parameter value 2.5 is m=2, and k-m-1=4-2-1=1.

- Continuity of position at parameter value 3, because the multiplicity of the parameter value 3 is m=3, and k-m-1=4-3-1=0

- Discontinuity at parameter value 4, because the multiplicity of the parameter value 4 is m=4, and k-m-1=4-4-1=-1.

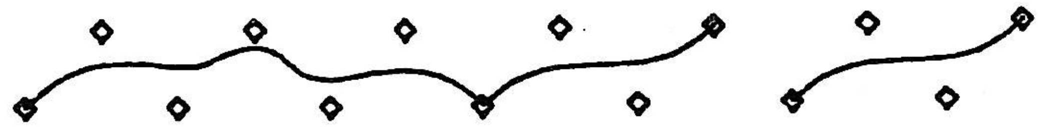

Figure 2.1. In addition to the vertices the knot values and multiplicities affect the shape of the curve.

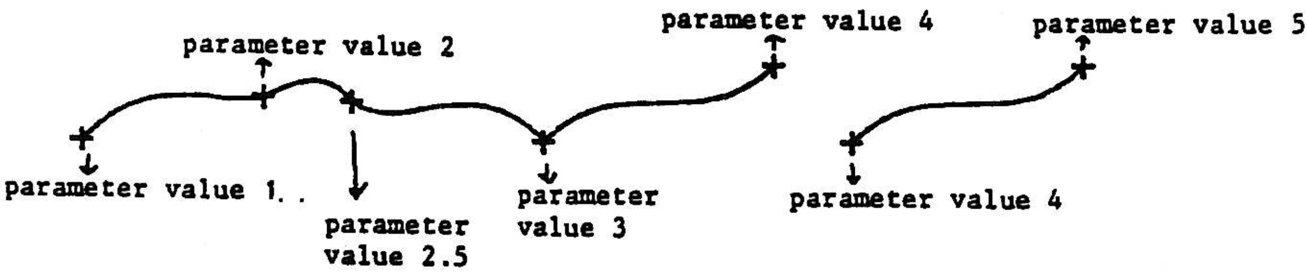

Figure 2.2. The multiplicities of the knots affect the continuity of the curve.

2.1.2 The B-spline format for surfaces

The B-spline description for surfaces is similar to the B-spline description for curves described in section 2.1.1. It consists of eight parts in stead of the five parts in the curve description.

t1(i)	- Knots in first parameter direction i=1,...,N1+K1
t2(i)	- Knots in second parameter direction i=1,...,N2+K2
C(l,i,j)	- Vertices of the B-spline surface, l=1,...,DIM, i=1,...,N1 and j=1,...,N2
N1	- Number of vertices in first parameter direction
N2	- Number of vertices in second parameter direction
K1	- Order in first parameter dirction
K2	- Order in second parameter direction
DIM	- The dimension of the space in which the surface lies.

The knot vectors describe the following properties of the surface:

- The parametrization of the B-spline surface
- The knot lines in the parameter plane of the surface will be lines along which some derivative of the surface is discontinuous.
- A rectangular area limited by knot lines, will be the domain of one of the polynomial pieces from which the surface is built up

The information of the knot vectors of a surface can be read in a similar way that we read the knot vector of a curve:

- The K1-th entry in the first knot vector is the parameter value of the left boundary of the surface, seen from the parameter plane of the surface.
- The (N1+1)-th entry in the first knot vector is the parameter value of the right boundary of the surface, seen from the parameter plane of the surface.
- The K2-th entry in the second knot vector is the parameter value of the lower boundary of the surface, seen from the parameter plane of the surface.
- The (N2+1)-th entry in the second knot vector is the parameter value of the upper boundary of the surface, seen from the parameter plane of the surface.
- The entries in the first knot vector with number K1+1,...,N1 are constant parameter lines where some derivative in the first parameter direction of the surface is discontinuous. Thus also a boundary between polynomial segments.
- The entries in the second knot vector with number K2+1,...,N2 are

constant parameter lines where some derivative in the second parameter direction of the surface is discontinuous. Thus also a boundary between polynomial segments.

- The knots are always listed in increasing order. More than one knot can have the same value. If some of the internal knots have the same value as the start or the end parameter value of the surface in the actual parameter direction, some of the description of the B-spline surface can be degenerate. Degenerate surface description do not affect the geometry algorithms.

- The continuity of the surface at the knot values depends on the number of knots having the same value. The number of knots having the same value is called the multiplicity, "m", of the knot value The order of continuity at a knot value is "k-m-1". ("k" is the order of the knot vector in question). If k knot values are equal, the surface is discontinuous at the constant parameter line in question. If the multiplicity "m"=1, the order of continuity across the parameter line is "k-2".

2.2 **Use of scratch array.**

To have dynamic allocation of memory, all space needed for arrays in the subroutines is allocated from a scratch array given as input to the subroutines. Which part of the scratch array is free for use, is indicated by two pointers. One pointer indicates the position of the first free element in the scratch array, while another pointer indicates the last free element in the scratch array. It is assumed that the space between the two pointers is free for use. The scratch needed is thus allocated from either the start of the free area or from the end of the free area.

The scratch allocated can be used in two ways:

- Internal use in the subroutines. This scratch is usually allocated from the end of the free scratch area, and is released before leaving the subroutine.

- Transport of output arrays from the subroutine to the program or subroutine calling the subroutine. This scratch is usually allocated from the start of the scratch array. Pointers into the scratch array will be output parameters from the subroutines. These pointers tell where the space allocated for the output is located in the scratch array. In addition information will be given telling the dimension of the scratch areas used for output.

NB! The scratch allocated for output from the subroutines can only be released by the program/subroutine calling the geometry routines. If scratch is not released, then at some stage in the calculation, all scratch can already have been allocated and an error status, "ISTAT=-3001", is returned from the geometry subroutine called.

This type of scratch allocation is possible since parameters to FORTRAN subroutines are transfered by a reference to the address of the parameter. Thus an actual call of a subroutine can be using a one dimensional array, while the subroutine can view the parameter as an array of higher dimension.

2.3 Use of parameter names.

The parameter names of the subroutines follow certain conventions. Arguments in different subroutines carrying the same type of information have been given the same names.

2.3.1 Parameter naming conventions.

The first character in parameter/variable names reflects the type of parameter/variable:

Integers

I - Single argument (Formal parameter)

J - Single common

K - Single local

L - Local array

M - Common array

N - Array argument

J - Function

Reals

A - Single argument

C - Single common

T - Single local (temporary)

S - Local array

G - Common array (global)

E - Array argument (external)

F - Function

2.?.2 Standard parameter names used for input parameters.

EPOINT - Array used for the input of points, tangents (derivatives) or second derivatives to subroutines producing curves. The dimension of the array is (1:IDIM,1:INBPNT).

INBPNT - An integer used to give the number of different points, tangents (derivatives) and second derivatives in the array EPOINT.

IDIM - The number of dimensions in the space in which the geometry we are working on, is lying.

EPTYP - An array used for describing the entries in the EPOINT array. For each entry it tells if the entry is a point, a tangent (derivative) or a second derivative. The dimension of the array is (1:INBPNT)

ICNSTA - Interpolation condition to be used at the start of a curve.

ICNEND - Interpolation condition to be used at the end of a curve.

CUOPEN - A logical parameter having the value .TRUE. if the curve to be produced is open and .FALSE. if the curve to be produced is closed.

ET - Array used for the input of knot values for curves. The dimension of ET is (1:IN+IK)

ET1 - Array used for the input of knot values in the first parameter direction of a surface. The dimension of ET1 is (1:IN1+IK1)

ET2 - Array used for the input of knot values in the second parameter direction of a surface. The dimension of ET2 is (1:IN2+IK2)

EBCOEF - Array used for the input of B-spline vertices/coefficients of a curve. The dimension of EBCOEF is (1:IDIM,1:IN).

ESURF - Array used for the input of B-spline vertices/coefficients of a surface. The dimension of ESURF is (1:IDIM,1:IN1,1:IN2)

IK - The polynomial order of the B-spline basis to be used for curves. The polynomial order of the B-spline basis is the polynomial degree plus 1.

IK1 - The polynomial order of the B-spline basis to be used in first parameter direction of a surface. The polynomial order of the B-spline basis is the polynomial degree plus 1.

IK2 - The polynomial order of the B-spline basis to be used in second parameter direction of a surface. The polynomial order of the B-spline basis is the polynomial degree plus 1.

IDERIV - Integer telling which derivative to calculate at a specified parameter value of a B-spline curve.

AEPSCO - The computation resolution. The relative accuracy wanted in the calculation. AEPSCO=10E-6, will give six digits. If this resolution is specified smaller than the computer resolution, then the computer resolution will be used in stead in the algorithms, since the algorithms detect when the computer resolution is reached.

AEPSGE - The resolution of the geometry. How close can two points be and avoid being treated as the same point.

2.3.3 Standard parameter names used for both input and output parameters.

ASTPAR - The parameter value to be used as the start parameter value of a piecewise polynomial curve. In this variable the parameter value of the end of the curve is returned when leaving the subroutine.

IN - The number of B-spline vertices/coefficients describing a B-spline curve. IN+IK is the number of knots describing a B-spline basis.

IN1 - The number of B-spline vertices/coefficients in first parameter direction of a surface. IN1+IK1 is the number of knots in the first parameter direction of a surface,

IN2 - The number of B-spline vertices/coefficients in second parameter direction of a surface. IN2+IK2 is the number of knots in the second parameter direction of a surface,

ESCR - The array to be used for allocation of scratch. Dimension (1:IMXSCR)

INXTFR - The first free element in the scratch array.

IMXSCR - The last free element in the scratch array.

2.3.4 Standard parameter names used for output parameters.

ISTAT - The status variable returned from the subroutine. In section 2.5. some standard values are given.

IKNT - Pointer into ESCR telling where the knots of a curve are stored. The knots occupy ESCR from entry IKNT up to and including entry IKNT+IN+IK-1.

IT1 - Pointer into ESCR telling where the knots calculated in first parameter direction of a surface are stored. The knots occupy ESCR from entry IT1 up to and including entry IT1+IN1+IK1-1.

IT2 - Pointer into ESCR telling where the knots calculated in second parameter direction of a surface are stored. The knots occupy ESCR from entry IT2 up to and including entry

IT2+IN2+IK2-1.

ICOEF - Pointer into ESCR telling where the B-spline coefficients/vertices of a curve or surface are stored. For curves the vertices occupy entry ICOEF up to and including entry ICOEF+IN*IDIM-1 in ESCR. For surfaces they occupy entry ICOEF up to and including entry ICOEF+IN1*IN2*IDIM-1 in ESCR.

IPAR - Pointer into ESCR telling where the parametrization of the points given as input in the array EPOINT(1:IDIM,1:INBPNT) are stored. The parameter values calculated are stored from entry IPAR up to and including entry IN+INBPAR-1 in ESCR. If multiple interpolation conditions are given at a specified point in EPOINT, i.e. a tangent or second derivative is given in addition to the point, only one parameter value is produced and stored in the indicated area of ESCR. Thus INBPAR can be smaller than INBPNT.

INBPAR - The number of parameter values calculated.

2.4 The sequence of parameters.

Some of the arguments used in more than one subroutine will always be used in a given sequence.

- The first of the parameters is always the status parameter ISTAT.
- When a B-spline curve is given as the input to a subroutine the sequence of the parameters describing the curve is always: ET, EBCOEF, IN, IK, IDIM.
- When a B-spline surface is given as the input to a subroutine the sequence of the parameters describing the surface is always: ET1, ET2, ESURF, IN1, IN2, IK1, IK2, IDIM;
- When a B-spline curve results from calculations in a subroutine, the pointers to the knots and vertices, and the numbers giving the number of vertices and the order will always be in the following order: IKNT, ICOEF, IN, IK.;
- When a B-spline surface results from calculations in a subroutine, the pointers to the knots and vertices, and the numbers giving the number of vertices and the order will always be in the following order: IT1, IT2, ICOEF, IN1, IN2, IK1, IK2.
- The parameters describing the scratch allocation will always be the three last parameters. Their sequence is: ESCR, INXTFR, IMXSCR.

2.5 Standard status variable values.

The status variable values returned form the subroutines are standardized. The values used are described in appendix A. the following conventions are followed for the error messages.

ISTAT < 0 - ERROR
ISTAT = 0 - OK
ISTAT > 0 - WARNING OR MESSAGE.

The two following status values are important to know. They will be among those appearing most frequently:

-9001 - Not enough scratch or error in scratch pointers.
Action: Check that INXTFR>0 and IMXSCR>0. If one of these are less than 1, the program calling the B-spline routines tries to reference array elements before the start of the scratch array. A correction must be done in the subroutine or program calling the B-spline subroutine that is giving the error status.
If INXTFR>0 and IMXSCR>0 the the ESCR array is to small, increase the size of the array and try once more.

-9003 - Error in B-spline description.
Action: Check the input curves or surfaces. Either the number of vertices is less than 1 or the number of vertices is less than the polynomial order of the curve/surface.

2.6 How to control the resolution factors

As mentioned in the list of standard parameter names the two resolution factors AEPSCO and AEPSGE exist. The way the values of these are set will both influence the accuracy and the execution time of the calculation subroutine. In the next two subsections some advice of the use of these factors will be given.

2.6.1 Control of AEPSCO, the computation resolution

The computation resolution is rather easy to control. The value of this should be specified in the interval between 1 and the computer resolution. If a too small value or a negative value is given, then the computer resolution is used. If we want the calculation to have five digits accuracy then AEPSCO=10E-5 is the value to use.

2.6.2 Control of AEPSGE, the geometry resolution

The geometry resolution specifies how close two points shall be to be regarded as the same point. AEPSGE specifies the maximum depth of the recursion in routines that use recursion. The geometry resolution should be given different values according to the kind of problem one intend to solve. The size of AEPSGE is related to the world coordinates of points and vertices.