11.6 Low-level Subprograms for Operations on Splines

A. Purpose

This chapter describes five subprograms for spline operations that are used by the subprograms of the preceding chapter. It is expected that one would only use these subprograms directly if one has needs more specialized than are covered by the higher level subprograms of the preceding chapter.

Subroutine DSVALA evaluates at an argument \( X \) the values of the derivatives, of orders 0 through \( \text{NDERIV} \), of a spline function represented using the B-spline basis. DSVALA must be given a difference table of the coefficients of the spline function. Subroutine DSDEF is provided to compute this difference table. Once the difference table has been computed and saved, use of DSVALA is more economical than making \( \text{NDERIV}+1 \) calls to subprogram DSVAL of the preceding chapter if \( \text{NDERIV} > 0 \).

Subroutine DSFIND does a lookup in a knot array to find a knot subinterval of nonzero length containing a specified argument \( X \), or the nearest such subinterval if extrapolation is needed.

Using a knot sequence regarded as defining a B-spline basis function of order \( \text{KORDER} \), subroutine DSBSAD computes the values at \( X \) of the \( \text{KORDER} \) B-spline basis functions (or a derivative of these functions as specified by \( \text{IDERIV} \)) that could be nonzero at \( X \). Subprogram DSBSAI computes the integral from \( X_1 \) to \( X_2 \) of each of the \( \text{NCOEF} \) basis functions. The output of these subprograms is needed in setting up the matrix for curve fitting or interpolation involving values, derivatives, or integrals of the fitted spline function.

B. Usage

B.1 Usage of DSBSAD for evaluation of basis functions or their derivatives

B.1.a Program Prototype, Double Precision

\[
\text{INTEGER KORDER, LEFT, IDERIV} \\
\text{DOUBLE PRECISION TKNOTS(} \geq \text{ncoef} + \text{KORDER}) \text{, X, BDERIV(} \geq \text{KORDER})
\]

(See TKNOTS below for the definition of \( \text{ncoef} \).)

Assign values to KORDER, LEFT, TKNOTS, X, and IDERIV.

\[
\text{CALL DSBSAD(KORDER, LEFT, TKNOTS, X, IDERIV, BDERIV)}
\]

Computed quantities are returned in BDERIV().

B.1.b Argument Definitions

**KORDER** [in] KORDER is both the order of the spline basis functions and the number of basis functions whose derivatives are to be evaluated.

**LEFT** [in] Identifies an interval of nonzero length [TKNOTS(LEFT), TKNOTS(LEFT+1)] which is the reference interval for the function evaluation. DSBSAD will evaluate the \( \text{IDERIV}^{th} \) derivative of the \( \text{KORDER} \) basis functions that could be nonzero on this interval. Require \( \text{KORDER} \leq \text{LEFT} \leq \text{ncoef} \). Except when extrapolation is needed, LEFT should satisfy TKNOTS(LEFT) \( \leq X < \text{TKNOTS(LEFT+1)} \). We recommend that the subroutine DSFIND be used to determine LEFT.

**TKNOTS()** [in] The knot sequence \( [t_i: i = 1, ..., \text{ncoef} + \text{KORDER}] \), where \( \text{ncoef} \) denotes the total number of B-spline basis functions associated with this knot sequence. The proper interpolation interval, \( [a, b] \), associated with this knot sequence is given by \( a = \text{TKNOTS(KORDER)} \) and \( b = \text{TKNOTS(ncoef+1)} \). Require \( t_i \leq t_{i+1} \) for \( i = 1, ..., \text{ncoef} + \text{KORDER} - 1; t_i < t_{i+\text{KORDER}} \) for \( i = 1, ..., \text{ncoef}; t_{\text{KORDER}+1} > t_{\text{KORDER}}; t_{\text{ncoef}+1} < t_{\text{ncoef}+1} \). The knots strictly between \( a \) and \( b \) are internal knots. They specify abscissas at which one polynomial piece ends and the next begins. Successive internal knots may have the same value. An abscissa appearing with multiplicity \( \mu \) means the order of continuity of the spline at this abscissa will be at least \( \text{KORDER} - \mu - 1 \). The knots indexed ahead of \( t_{\text{KORDER}} \) can all be equal to \( a \), and those indexed after \( t_{\text{ncoef}+1} \) can all be equal to \( b \).

**X** [in] Argument at which the \( \text{IDERIV}^{th} \) derivative of basis functions are to be evaluated.

**IDERIV** [in] Order of derivative to be computed. IDERIV = 0 specifies function values. Require IDERIV \( \geq 0 \). Values of derivatives of order \( \geq \text{KORDER} \) will be zero.

**BDERIV()** [out] On return the values at \( X \) of the \( \text{IDERIV}^{th} \) derivative of the basis functions indexed from LEFT + 1 - \( \text{KORDER} \) through LEFT will be stored in BDERIV(i), \( i = 1, ..., \text{KORDER} \).

B.2 Usage of DSBSAI for evaluation of an integral of basis functions

B.2.a Program Prototype, Double Precision

\[
\text{INTEGER NCOEF, J1, J2} \\
\text{DOUBLE PRECISION TKNOTS(} \geq \text{NCOEF} + \text{KORDER}) \text{, X1, X2, BASI(} \geq \text{NCOEF})
\]
Assign values to KORDER, NCOEF, TKNOTS(), X1, X2, J1, and J2.

CALL DSBASI ( KORDER, NCOEF, TKNOTS, X1, X2, J1, J2, BASI)

Computed results are returned in J1, J2, and BASI().

B.2.b Argument Definitions

KORDER [in] The order of the spline basis functions.

NCOEF [in] The total number of B-spline basis functions associated with this knot sequence. Also the number of values to be returned in BASI().

TKNOTS() [in] As for DSBASD above, with ncoef replaced by NCOEF.

X1, X2 [in] Integration is to be done from X1 to X2. Permit X1 < X2 or X1 ≥ X2. Generally X1 and X2 should each lie in [a, b], however extrapolation will be used to return values when this is not the case.

J1, J2 [inout] On entry J1 and J2 must contain integer values. If J1 is in [1, N] it will be used to start the lookup for X1. Otherwise the search will start with 1. Similarly for J2.

On return J1 and J2 indicate the portion of the array BASI() that might be nonzero on return. BASI(i) might be nonzero if J1 ≤ i ≤ J2, and BASI(i) = 0 if i < J1 or i > J2.

BASI() [out] On return, BASI(i) will contain the value of the integral of the ith basis function over the range from X1 to X2, for i = 1, ..., NCOEF. J1 and J2 above indicate which elements might be nonzero.

B.3 Usage of DSDIF to compute the difference table needed by DSVALA

B.3.a Program Prototype, Double Precision

INTEGER KORDER, NCOEF, NDERIV

DOUBLE PRECISION TKNOTS(≥ NCOEF + KORDER), BCOEF(≥ NCOEF),
BDIF(≥ NCOEF × (NDERIV+1))

Assign values to KORDER, NCOEF, TKNOTS(), BCOEF(), and NDERIV.

CALL DSDIF (KORDER, NCOEF, TKNOTS, BCOEF, NDERIV, BDIF)

Computed results are returned in BDIF().

B.3.b Argument Definitions

KORDER [in] The order of the spline basis functions.

NCOEF [in] The total number of B-spline basis functions associated with this knot sequence.

TKNOTS() [in] Same specifications as for DSBASI above.

BCOEF() [in] Array of NCOEF coefficients representing a spline function relative to a B-spline basis.

NDERIV [in] Highest order difference to be computed. Since the difference table BDIF() is intended for use by DSVALA this should correspond to the largest order derivative one intends to compute using DSVALA.

BDIF() [out] Will contain a copy of BCOEF() plus differences through order NDERIV of this array of coefficients. Intended for use by DSVALA.

B.4 Usage of DSFIND for lookup in a knot sequence

B.4.a Program Prototype, Double Precision

INTEGER IX1, IX2, LEFT, MODE

DOUBLE PRECISION XT(IX2+1), X

Assign values to XT(), IX1, IX2, LEFT, and X.

CALL DSFIND(XT, IX1, IX2, X, LEFT, MODE)

Results are returned in LEFT and MODE.

B.4.b Argument Definitions

XT(), IX1, IX2 [in] XT() is the array in which the lookup will be done. DSFIND will only look at elements from XT(IX1) through XT(IX2). Require IX1 < IX2, XT(IX1) < XT(IX1+1), XT(IX2−1) < XT(IX2), and XT(i) ≤ XT(i+1) for i = IX1+1, ..., IX2 − 2.

If the lookup is in a knot array of length korder + ncoef associated with a B-spline basis, one would generally set IX1 = korder and IX2 = ncoef + 1. If the lookup is in a knot array of length npc + 1 associated with a power basis, one would generally set IX1 = 1 and IX2 = npc + 1.

X [in] Value to be looked up in XT().

LEFT [inout] On entry LEFT must contain an integer value. If this value is in [IX1, IX2 − 1] the lookup will start with this value, otherwise the lookup starts with IX1 or IX2 − 1.

On return LEFT is the index of the left end of the reference interval ⟨ XT(LEFT), XT(LEFT+1) ⟩ for X.
This will always be an interval of nonzero length. If $X$ satisfies $XT(IX1) \leq X < XT(IX2)$ then LEFT will satisfy $XT(LEFT) \leq X < XT(LEFT+1)$. Otherwise, if $X < XT(IX1)$, $LEFT = IX1$; or if $X \geq XT(IX2)$, $LEFT = IX2 - 1$. The polynomial segment defined over this reference interval is intended to be used for function evaluation at $X$.

**MODE** [out] Indicator of the position of $X$ relative to $[XT(IX1), XT(IX2)]$. Set to $-1$ if $X$ is to the left of this interval, to $0$ if $X$ is in this closed interval, and to $+1$ if $X$ is to the right of this interval.

**B.5 Usage of DSVALA for evaluating a sequence of derivatives**

**B.5.a Program Prototype, Double Precision**

```
INTEGER KORDER, NCOEF, NDERIV
DOUBLE PRECISION TKNOTS(\geq NCOEF + KORDER), BDIF(\geq NCOEF \times (NDERIV+1)), X, SVALUE(\geq NDERIV + 1)
```

Assign values to KORDER, NCOEF, TKNOTS(), NDERIV, BDIF(), and X.

```
CALL DSVALA(KORDER, NCOEF, TKNOTS, NDERIV, BDIF, X, SVALUE)
```

Computed results are returned in SVALUE().

**B.5.b Argument Definitions**

**KORDER** [in] The order of the spline basis functions.

**NCOEF** [in] The total number of B-spline basis functions associated with this knot sequence.

**TKNOTS()** [in] Same specifications as for DSBSAI above.

**NDERIV** [in] Highest order derivative to be evaluated. Values of derivatives of order $\geq$ KORDER will be zero.

**BDIF()** [in] A difference table of B-spline coefficients computed by DSDFIF.

**X** [in] Argument at which values returned in SVALUE() are to be computed.

**SVALUE()** [out] On return, SVALUE($i+1$) contains the value at $X$ of the $i^{th}$ derivative of the spline function $f$ for $i = 0, ..., NDERIV$. The spline function $f$ is defined by the parameters KORDER, NCOEF, TKNOTS(), and the coefficients whose difference table is in BDIF().

**B.6 Modifications for Single Precision**

For single precision usage change the DOUBLE PRECISION statements to REAL and change the initial “D” in the subprogram names to “S”.

**C. Examples and Remarks**

The program DRDSBASD and output listing ODDSBSAI demonstrate the use of the subprograms of this chapter.

**D. Functional Description**

The subprograms of this chapter are described in Section D of Chapter 11.5.

**References**


**E. Error Procedures and Restrictions**

DSBSAI, DSBSAI, and DSVALA each contain an internal dimensioning parameter $kmax = 20$. It is an error if KORDER $> kmax$ in DSBSAI, DSBSAI, or DSVALA. The condition IDERIV $< 0$ is an error in DSBSAI.

In DSFIND, if the search reaches either of the intervals $[XT(IX1), XT(IX1+1)]$ or $[XT(IX2−1), XT(IX2)]$ and the interval is found to have nonpositive length, the error is reported.

Errors are reported to the library error message processing subroutines of Chapter 19.2 with a severity level of 2 that will, by default, cause execution of the program to stop.

Abscissae and weights for 2-point, 6-point, and 10-point Gaussian quadrature are stored to 40 decimal digits in DSBSAI. With infinite precision abscissae and weights, these formulae would be exact for splines of KORDER up to 20.

**F. Supporting Information**

The source language is ANSI Fortran 77.

These subprograms, except DSBSAI, are modifications by Lawson of codes developed by C. de Boor, [1]. Subprogram DSBSAI is based on code due to D. E. Amos, [2].
DRDSBASD

program DRDSBASD
C 1996-07-09 DRDSBASD Krogh Format changes for conversion to C.
C 1994-10-19 DRDSBASD Krogh Changes to use M7MC
C 1993-01-12 DRDSBASD CLL @ JPL
C 1992-11-12 CLL @ JPL
C Demo driver for DSBASD, DSBASI, DSDIF, DSFIND, DSVALA.
C
C
integer I, IDERIV, J1, J2, KORDER, LEFT, MODE, NCOEF, NDERIV
parameter(KORDER = 4, NCOEF = 6)
parameter( IDERIV = 2, NDERIV = 3)
double precision BASI(NCOEF), BCOEF(NCOEF), BDERIV(KORDER)
double precision BDIF(NCOEF*NDERIV+1), BVALS(KORDER)
double precision SVALUE(NDERIV+1)
double precision TKNOTS(KORDER+NCOEF), X, X1, X2
parameter( X = 0.4d0, X1 = 0.1d0, X2 = 0.9d0)
data TKNOTS / 4*0.0d0, 0.3d0, 0.8d0, 4*1.0d0 /
data BCOEF / 0.1d0, 0.4d0, 0.2d0, -0.3d0, -0.5d0, -0.2d0 /

C print('( ' 'DRDSBASD.. Demo driver for' ')/
* ' 'DSBASD, DSBASI, DSDIF, DSFIND, DSVALA. ' '/
* ' 'KORDER =', i2, ', NCOEF =', i2, ' TKNOTS() =', v(1),
* ' 10*5.1', ' KORDER, NCOEF, (TKNOTS(1), I=1,KORDER+NCOEF)
print('( ' 'Using DSFIND with X = ', f5.1), X
LEFT = 1
call DSFIND(TKNOTS, KORDER, NCOEF+1, X, LEFT, MODE)
print('( ' 'X: ', i2/6x, 4f12.6), ' LEFT+1-KORDER, LEFT,
* (BVALUES(I), I=1,KORDER)
print('( ' 'Using DSBASD with IDERIV = 0'),
call DSBASD(KORDER, LEFT, TKNOTS, X, 0, BVALUES)
print('(3x,', ' Values at X of basis functions indexed from ', i2, 
* ' to ', i2/6x, 4f12.6), ' LEFT+1-KORDER, LEFT,
* (BVALUES(I), I=1,KORDER)
print('( ' 'Using DSBASD with IDERIV = ', i2), IDERIV
call DSBASD(KORDER, LEFT, TKNOTS, X, IDERIV, BDERIV)
print('(3x,
* ' 'Values at X of 2nd deriv of basis functions indexed from ',
* i2, ' to ', i2/6x, 4f12.5), ' LEFT+1-KORDER, LEFT,
* (BDERIV(I), I=1,KORDER)
print('( ' 'Using DSBSA with X1 = ', f5.1, ' and X2 = ', f5.1),
* X1, X2
J1 = 1
J2 = 1
call DSBASI(KORDER, NCOEF, TKNOTS, X1, X2, J1, J2, BASI)
print('(3x,', ' J1 =', i2, ',', ' J2 =', i2/3x,
* Integral from $X_1$ to $X_2$ of basis functions: $'/3x, 6f11.6')$,
* $J_1, J_2, (BASI(1), I=1:NCOEF)$
* print '(Using DSDIF and DSVALA with NDERIV =''i2'', and''/3x,''
* BCOEF() = ''6f5.1)'', NDERIV, (BCOEF(I), I=1:NCOEF)
* call DSDIF(KORDER, NCOEF, TNKNOTS, BCOEF, NDERIV, BDIF)
* call DSVALA(KORDER, NCOEF, TNKNOTS, NDERIV, BDIF, X, SVALUE)
* print 'Values of derivs 0 through ''i2'' at $X$: ''/
* (3x, 'Values of derivs 0 through ''i2'' at $X$: ''/
* 6x,4f11.6)'', NDERIV, (SVALUE(I), I=1,NDERIV+1)
* end

**ODDSBASD**

DRDSBASD.. Demo driver for
DSBASE, DSBASEI, DSDIF, DSFIND, DSVALA.
KORDER = 4, NCOEF = 6
TNKNOTS() = 0.0 0.0 0.0 0.0 0.0 0.3 0.8 1.0 1.0 1.0 1.0
Using DSFIND with $X = 0.4$
LEFT = 5, MODE = 0
Using DSBASD with IDERIV = 0
Values at $X$ of basis functions indexed from 2 to 5
0.200000 0.542857 0.253061 0.004082
Using DSBASD with IDERIV = 2
Values at $X$ of 2nd deriv of basis functions indexed from 2 to 5
7.500000 −11.78571 1.83673 2.44898
Using DSBAISI with $X1 = 0.1$ and $X2 = 0.9$
J1 = 1, J2 = 6
Integrals from $X_1$ to $X_2$ of basis functions:
0.014815 0.163874 0.246236 0.244080 0.127870 0.003125
Using DSDIF and DSVALA with NDERIV = 3 and
BCOEF() = 0.1 0.4 0.2 −0.3 −0.5 −0.2
Values of derivs 0 through 3 at $X$:
0.110612 −1.181633 −1.132653 7.423469

July 11, 2015 Low-level Subprograms for Operations on Splines 11.6–5