11.2 Evaluation, Integration, and Differentiation of Polynomials

A. Purpose

This set of subroutines will evaluate, integrate, or differentiate polynomials. The polynomials may be represented by coefficients relative to either the monomial or Chebyshev basis. The data structure and parameterization used to represent a polynomial is the same as that used by the least-squares polynomial curve fit subroutines described in Chapter 11.1. Special procedures for evaluation of polynomials expressed using the Legendre and Laguerre bases are described in Chapters 2.11 and 2.12 respectively.

B. Usage

B.1 Usage for Evaluation

B.1.a Program Prototype, Single Precision

INTEGER NDEG

REAL P(≥NDEG+3)**, X, Y, SCPVAL, SMPVAL**

Assign values to NDEG, X, and P(i), $i = 1$, NDEG+3. If the Chebyshev basis is being used, use the statement:

Y = SCPVAL(P, NDEG, X)

If the monomial basis is being used, use the statement:

Y = SMPVAL(P, NDEG, X)

Following the appropriate one of these two statements the value of the polynomial at the argument X will be stored in Y.

B.1.b Argument Definitions

P() [in] An array containing NDEG+3 parameters that define a polynomial as described in Section D.

NDEG [in] Degree of the polynomial.

X [in] Argument value at which the polynomial is to be evaluated.

SCPVAL [out] The value of the polynomial evaluated at X assuming the Chebyshev basis representation.

SMPVAL [out] The value of the polynomial evaluated at X assuming the monomial basis representation.

B.2 Usage for Integration

B.2.a Program Prototype, Single Precision

INTEGER NDEGA, NDEGB

REAL A(≥NDEGA+3)**, B**(≥NDEGA+4)

Assign values to NDEGA and A(i), i = 1, ..., NDEGA+3. If the Chebyshev basis is being used, use the statement:

CALL SCPINT (A, NDEGA, B, NDEGB)

If the monomial basis is being used, use the statement:

CALL SMPINT (A, NDEGA, B, NDEGB)

Following the appropriate one of these two call statements the results will be stored in B() and NDEGB.

B.2.b Argument Definitions

A() [in] An array containing NDEGA+3 parameters that define the input polynomial, say $p(x)$. See Section D for the specification of the parameterization.

NDEGA [in] Degree of the input polynomial, $p(x)$.

B() [out] On return B() will contain NDEGB+3 parameters defining the output polynomial, say $q(x)$, which is the indefinite integral of the input polynomial $p(x)$. Mathematically the constant term of $q(x)$ is an arbitrary constant of integration. This subroutine will set the constant term, B(3), to zero. The storage locations occupied by A() and B() must be distinct.

NDEGB [out] The subroutine sets NDEGB = NDEGA+1 to indicate the degree of the output polynomial. The storage locations occupied by NDEGA and NDEGB must be distinct.

B.3 Usage for Differentiation

B.3.a Program Prototype, Single Precision

INTEGER NDEGC, NDEGD

REAL C (≥NDEGC+3)**, D**($\geq \max(3, \mathrm{NDEGC} + 2)$)

Assign values to NDEGC and C(i), i = 1, ..., NDEGC+3. If the Chebyshev basis is being used, use the statement:

CALL SCPDRV (C, NDEGC, D, NDEGD)

If the monomial basis is being used, use the statement:

CALL SMPDRV (C, NDEGC, D, NDEGD)

Following the appropriate one of these two call statements the results will be stored in D() and NDEGD.

B.3.b Argument Definitions

C() [in] An array containing NDEGC+3 parameters that define the input polynomial, say $p(x)$. See Section D for the specification of the parameterization.

NDEGC [in] Degree of the input polynomial, $p(x)$.

D() [out] On return D() will contain NDEGD+3 parameters defining the output polynomial, say $q(x)$, which is the derivative of the input polynomial $p(x)$. The storage locations occupied by C() and D() must be distinct.

NDEGD [out] The subroutine sets NDEGD = max(0, NDEGC−1) to indicate the degree of the output polynomial. The storage locations occupied by NDEGC and NDEGD must be distinct.

B.4 Usage for Double Precision Evaluation, Integration or Differentiation

For DOUBLE PRECISION usage change the REAL type statements to DOUBLE PRECISION and change the initial "S" of the function and subroutine names to a "D." Note particularly that if the function names DCPVAL or DMPVAL are used they must be typed DOUBLE PRECISION either explicitly or via an IMPLICIT statement.

C. Examples and remarks

Let a cubic polynomial $p(x)$ be defined relative to the Chebyshev basis as $p(x) = 10+8T_1(u)+6T_2(u)+4T_3(u)$ where $u = (x - 5)/2$. The DRSCPVAL program computes the indefinite integral of $p(x)$ calling it $q(x)$. This computation is checked by computing $r(x)$ as the derivative of $q(x)$. Note that $r(x)$ agrees with $p(x)$. Finally the program evaluates the definite integral

$$z = \int_4^6 p(x)\,dx = q(6) - q(4) = 10$$

The output from this program is shown in ODSCPVAL.

D. Functional Description

In typical expected usage the polynomial parameter vector input to any of the subprograms of this set will have been produced by the library curve fitting subroutine SPFIT (or DPFIT) or an integration or differentiation subroutine of this set. The subprograms of this set are thus intended to let the user do the operations of evaluation, integration or differentiation of polynomials without being concerned with the details of the parametric representation of polynomials or algorithmic details.

The following description is provided for those who wish to know more of the internal details.

For the purposes of this set of subprograms a polynomial of degree n, say $p(x)$, is represented by a set of $n+3$ parameters, say a_1, ..., a_{n+3}. The first two parameters define a linear transformation of the independent variable

$$u = (x - a_1)/a_2$$

The remaining $n+1$ parameters are coefficients of an n^{th} degree polynomial in the transformed variable u. If the Chebyshev basis is used this polynomial is

$$p(x) = q(u) = \sum_{i=0}^{n} a_{i+3} T_i(u)$$

whereas if the monomial basis is used the polynomial is

$$p(x) = q(u) = \sum_{i=0}^{n} a_{i+3} u^i$$

The Chebyshev polynomials $T_i(u)$ are defined by the equations

$$T_0(u) = 1, \quad T_1(u) = u$$
$$T_i(u) = 2uT_{i-1}(u) - T_{i-2}(u), \quad i = 2, 3, ...$$

The formulas for differentiation and integration of polynomials expressed using the Chebyshev basis may be derived from the following standard identities:

$$0 = dT_0(u)/du$$
$$T_0(u) = dT_1(u)/du$$
$$T_1(u) = \frac{1}{4} dT_2(u)/du$$
$$T_i(u) = \frac{1}{2}\frac{d}{du}\left[\frac{T_{i+1}(u)}{i+1} - \frac{T_{i-1}(u)}{i-1}\right], \quad i = 2, 3, ...$$

The algorithms used by this set of subprograms are specified as follows:

D.1 Monomial Basis Evaluation, SMPVAL or DMPVAL

Given an n^{th} degree polynomial p represented by the parameters p_i, $i = 1$, ..., $n+3$ and an argument x, compute $y = p(x)$.

$$u = (x - p_1)/p_2$$
$$z_n = p_{n+3}$$
$$z_i = u\, z_{i+1} + p_{i+3}, \quad i = n-1, n-2, ..., 0$$
$$y = z_0.$$

D.2 Monomial Basis Integration, SMPINT or DMPINT

Given an n^{th} degree polynomial p represented by the parameters a_i, $i = 1$, ..., $n+3$, compute the parameters b_i that represent a polynomial q that for arbitrary u and v satisfies

$$\int_u^v p(x)\,dx = q(v) - q(u).$$

The formulas used are

$$b_1 = a_1, \quad b_2 = a_2, \quad b_3 = 0$$
$$b_{i+3} = a_2 a_{i+2}/i, \quad i = 1, ..., n+1$$

D.3 Monomial Basis Differentiation, SMPDRV or DMPDRV

Given an n^{th} degree polynomial p represented by the parameters c_i, $i = 1, ..., n+3$, compute the parameters d_i that represent the polynomial q satisfying

$$\frac{d}{dx}p(x) = q(x).$$

The formulas used are

$$d_1 = c_1, \quad d_2 = c_2$$
$$d_{i+3} = (i+1)c_{i+4}/c_2, \quad i = 0, ..., n-1$$

with a special case of $d_3 = 0$ if $n = 0$.

D.4 Chebyshev Basis Evaluation, SCPVAL or DCPVAL

Given an n^{th} degree polynomial p represented by the parameters p_i, $i = 1, ..., n+3$ and an argument x compute $y = p(x)$.

$$u = (x - p_1)/p_2$$
$$z_n = p_{n+3}$$
$$z_{n-1} = 2uz_n + p_{n+2}$$
$$z_i = 2uz_{i+1} - z_{i+2} + p_{i+3}, \quad i = n-2, ..., 1$$
$$y = uz_1 - z_2 + p_3$$

D.5 Chebyshev Basis Integration, SCPINT or DCPINT

Given an n^{th} degree polynomial p represented by the parameters a_i, $i = 1,..., n+3$, compute the parameters b_i that represent a polynomial q that for arbitrary u and v satisfies

$$\int_u^v p(x)\,dx = q(v) - q(u).$$

The formulas used are

$$b_1 = a_1, \quad b_2 = a_2, \quad b_3 = 0$$
$$b_4 = a_2\,[a_3 - (1/2)a_5]$$
$$b_{i+3} = a_2(a_{i+2} - a_{i+4})/(2i), \quad i = 2, ..., n+1,$$

where a_i for $i > n+3$ is taken to be zero.

D.6 Chebyshev Basis Differentiation, SCPDRV or DCPDRV

Given an n^{th} degree polynomial p represented by the parameters c_i, $i = 1, ..., n+3$, compute the parameters d_i that represent the polynomial q satisfying

$$\frac{d}{dx}p(x) = q(x)$$

The formulas used are

$$d_1 = c_1, \quad d_2 = c_2$$
$$d_{i+3} = 2(i+1)c_{i+4}/c_2, \quad i = n-1, n-2$$
$$d_{i+3} = d_{i+5} + 2(i+1)c_{i+4}/c_2, \quad i = n-3, ..., 1$$
$$d_3 = d_5/2 + c_4/c_2$$

with a special case of $d_3 = 0$ if $n = 0$.

E. Error Procedures and Restrictions

The degree of the input polynomial must be zero or positive. If it is negative the subprograms in this set will issue an error message using the error processor in Chapter 19.2 at level 0 and return.

The given values of P(2), A(2), and C(2) must be nonzero. These conditions are not tested.

The storage locations for the input quantities in the integration and differentiation subroutines must be distinct from the storage locations for the output quantities.

Since DCPVAL and DMPVAL are FUNCTION type subprograms, their names must be typed DOUBLE PRECISION in any program that uses them.

F. Supporting Information

Entry	Required Files
DCPDRV	DCPDRV, ERFIN, ERMSG, IERM1, IERV1
DCPINT	DCPINT, ERFIN, ERMSG, IERM1, IERV1
DCPVAL	DCPVAL
DMPDRV	DMPDRV, ERFIN, ERMSG, IERM1, IERV1
DMPINT	DMPINT, ERFIN, ERMSG, IERM1, IERV1
DMPVAL	DMPVAL
SCPDRV	ERFIN, ERMSG, IERM1, IERV1, SCPDRV
SCPINT	ERFIN, ERMSG, IERM1, IERV1, SCPINT
SCPVAL	SCPVAL
SMPDRV	ERFIN, ERMSG, IERM1, IERV1, SMPDRV
SMPINT	ERFIN, ERMSG, IERM1, IERV1, SMPINT
SMPVAL	SMPVAL

The source language is Fortran 77.

Designed by C. L. Lawson, JPL, 1970. Programmed by Lawson and D. Campbell, JPL, 1970. Adapted to Fortran 77 by Lawson and S. Y. Chiu, JPL, 1984.

DRSCPVAL

```
c       DEMONSTRATE SCPDRV, SCPINT, AND SCPVAL.
c>> 2001-05-22 DRSCPVAL Krogh Minor change for making .f90 version.
c>> 1997-05-29 DRSCPVAL Krogh Special code for C conversion.
c>> 1996-05-28 DRSCPVAL Krogh Added external statement.
c>> 1994-10-19 DRSCPVAL Krogh  Changes to use M77CON
c>> 1987-12-09 DRSCPVAL Lawson  Initial Code.
c--S replaces "?": DR?CPVAL, ?CPDRV, ?CPINT, ?CPVAL
c
      external SCPVAL
c++ Code for .C. is inactive
c%%   long int i;
c++ End
      integer NQ, NR
      real                 P(6),Q(7),R(6),Z,SCPVAL
      data       P/ 5.E0, 2.E0, 10.E0, 8.E0, 6.E0, 4.E0 /
c
      call SCPINT(P,3,Q,NQ)
      call SCPDRV(Q,NQ,R,NR)
      Z=SCPVAL(Q,NQ,6.E0)-SCPVAL(Q,NQ,4.E0)
c++ Code for ~.C. is active
      write(*,1000) P,Q,R,Z
 1000 format(21X,'P =',2F4.0,2X,4F7.2//' INTEGRAL OF P.        Q ='
     *,2F4.0,2X,5F7.2//' DERIVATIVE OF Q.     R =',2F4.0,2X,4F7.2/
     */' DEFINITE INTEGRAL.  Z =',F20.8)
c++ Code for .C. is inactive
c%%   printf( "                     P =%4.0f%4.0f   ", p[0], p[1]);
c%%   for(i=2L; i < sizeof(p)/sizeof(p[1]); i++)
c%%           printf( "%7.2f", p[i] );
c%%   printf( "\n INTEGRAL OF P.        Q =%4.0f%4.0f   ", q[0], q[1]);
c%%   for(i=2L; i < sizeof(q)/sizeof(q[1]); i++)
c%%           printf( "%7.2f", q[i] );
c%%   printf( "\n DERIVATIVE OF Q.     R =%4.0f%4.0f   ", r[0], r[1]);
c%%   for(i=2L; i < sizeof(r)/sizeof(r[1]); i++)
c%%           printf( "%7.2f", r[i] );
c%%   printf( "\n DEFINITE INTEGRAL.  Z =%7.2f\n", z );
c++ End
      stop
      end
```

ODSCPVAL

```
                     P =  5.  2.   10.00   8.00   6.00   4.00

 INTEGRAL OF P.      Q =  5.  2.    0.00  14.00   2.00   2.00   1.00

 DERIVATIVE OF Q.    R =  5.  2.   10.00   8.00   6.00   4.00

 DEFINITE INTEGRAL.  Z =          10.00000000
```