Algorithms L.D. Fosdick and A.K. Cline, Editors

Algorithm 446

Ten Subroutines for the Manipulation of Chebyshev Series [C1]

R. Broucke [Recd. 17 May 1971 and 7 April 1972]
University of California, Los Angeles, CA 90024, and
Jet Propulsion Laboratory, Pasadena, Calif.

Key Words and Phrases: Chebyshev series, differentiation, integration, curve fitting, approximations, negative powers
CR Categories: 5.12, 5.13, 5.16
Language: Fortran

Description

Introduction. These subroutines deal with the manipulation of Chebyshev series. The operations performed are the construction of the Chebyshev approximation of functions, the evaluation of the series or their derivative, the integration or differentiation, and the construction of negative or fractional powers of such a series.

The subroutines are written in ANSI Fortran. They have been used without modification on such computers as the IBM-7094, IBM-360/91 (Fortran-IV-G compiler) and Univac 1108 (Fortran-V compiler).

The ten subroutines are considered as a single set, principally because they all use the same storage philosophy. All information is transmitted through the *CALL*-sequence rather than through the use of *COMMON* statements. Therefore, the user must provide storage for all the series in his main program, taking into account that all operations are performed in double precision. The coefficients of each series occupy a one-dimensional double-precision array according to the rules of ANSI Fortran. When several Chebyshev series are being manipulated, it is convenient to store all the series in a matrix. Each column of the matrix contains a single series, in order that the coefficients of each series occupy consecutive storage locations.

The first six subroutines contain no calls to other subroutines; in this sense they may be considered as independent. Each subroutine can be used separately.

In the present type of operations, it is extremely important to design and perform a large number of tests to certify all of the subroutines. We have tested the subroutines by generating some Chebyshev series which were published by Clenshaw [4], but we have also tested them with a number of additional methods; for instance:

a. The series for several elementary functions such as $\sin(x)$, $\cos(x)$, $\sin(2x)$, and $\cos(2x)$ have been constructed directly. These series have then been evaluated, and the values have been compared with the values of the functions.

b. The series for $\cos(2x)$ and $\sin(2x)$ have been derived from the series $\sin(x)$ and $\cos(x)$ by multiplication and addition of series.

c. The series for $\sin(x)$ and $\cos(x)$ have been derived from each other by integration and differentiation.

d. Many tests have been made by multiplying a series $f(x)$ by the series $1/f(x)$ or for instance by squaring the series for $f(x)^{\frac{1}{2}}$, or other similar operations.

The generation, evaluation and multiplication subroutines. The methods for the generation of a Chebyshev series have been taken from C.W. Clenshaw's papers [3, 4, 5]. The rule for the multiplication of Chebyshev series is also described by Clenshaw [3, p. 137], but the flowchart of our subroutine is from L. Carpenter [2].

We only consider the interval $(-1, +1)$ of the independent variable x, and we represent a truncated Chebyshev series of order n in the form:

$$f(x) = (c_0/2) + c_1T_1(x) + c_2T_2(x) + \cdots + c_nT_n(x). \tag{1}$$

We want to draw the user's attention to the fact that we use a factor $\frac{1}{2}$ in the zero-order term but not in the last term of the series. Some authors have used different conventions in relation to this factor $\frac{1}{2}$ for the first and last terms.

In the applications of the subroutines some caution is also necessary, because the independent variable x (the Chebyshev independent variable) is within the limits $(-1, +1)$. If the user's variable t (the physical independent variable) is within the limits (t_1, t_2), the conversions between t and x should be made with the linear relations

$$t = ((t_2 + t_1)/2) + ((t_2 - t_1)/2)x; \qquad x = ((2t - (t_2 + t_1))/(t_2 - t_1)). \tag{2}$$

The coefficients c_i in formula (1) are computed with the rule given by Clenshaw [4, p. 3]:

$$c_i = (2/n \sum_{j=0}^{n}{}'' f(\cos(\pi j/n)) \cos(\pi ij/n); \qquad i = 0, 1, \ldots, n. \tag{3}$$

The double accent means that the first and last terms of the sum are divided by two. It is seen that $n + 1$ special values of the function $f(x)$ are needed. In some applications, n has been as large as 1,500.

A large number of applications have shown that in most instances the user desires to construct the Chebyshev series for not just one function but for several functions simultaneously. For instance, in the study of the motion of a particle there will always be three coordinates, x_1, x_2, x_3, rather than just one. For this reason we programmed the subroutine *CHEBY* to efficiently construct several Chebyshev series simultaneously. In particular, the number of cosine calculations has been minimized. There will be only $2n$ cosine calculations, no matter how many functions are being analyzed simultaneously.

Besides the main program, the user will have to provide his own subroutine for the evaluation of the special values of the functions to be analyzed, as explained in the comments of the subroutine *CHEBY*. The user may choose any name for this subroutine; however, this name has to be transmitted through the *CALL CHEBY*-statement. This function subroutine will generally evaluate the function values either by using the appropriate formulas or by performing table lookup and interpolations if the data is only available in the form of a table with discrete points.

The subroutine *ECHEB* evaluates a Chebyshev series with the aid of Clenshaw's recurrence rule [4, p. 9]. The c_i's being the coefficients of the given series, we compute the values $b_{n+2}, b_{n+1}, b_n, \ldots, b_0$ with:

$$b_{n+2} = b_{n+1} = 0; \qquad b_i = 2xb_{i+1} - b_{i+2} + c_i, \tag{4}$$

where the subscript i runs from n to 0. The number of arithmetic operations involved is only $3n$, and the value of the function is then $f(x) = (b_0 - b_2)/2$.

The subroutine *EDCHB* evaluates the derivative of a Chebyshev series (without storing the coefficients of the differentiated series). It implements a combination of the evaluation formula (4) and the differentiation formula (6) given below.

The differentiation and integration subroutines. Clenshaw's formulas [4, p. 11] have again been used for the differentiation and integration operations. The coefficients a_i of the integrated Chebyshev series are derived from the input coefficients c_i by:

$$a_0 = 0; \quad a_n = c_{n-1}/2n; \quad a_i = (c_{i-1} - c_{i+1})/2i; \quad i = 1, 2, \ldots, n-1. \tag{5}$$

The coefficients d_i of the differentiated series are obtained by a set of recurrence equations:

$$d_n = 0; \quad d_{n-1} = 2nc_n; \quad d_{i-1} = d_{i+1} + 2ic_i; \quad i = n-1, n-2, \ldots, 1. \tag{6}$$

When using the differentiation and integration subroutines, the user should remember the relation between the differentials of t and x:

$$dt = ((t_2 - t_1)/2)\, dx = (\Delta t/2) dx. \tag{7}$$

This should be considered whenever differentiation or integration of Chebyshev series is performed. For instance we have for any Chebyshev series f:

$$\int f\, dt = (\Delta t/2) \int f\, dx. \tag{8}$$

Negative and fractional powers. Our last four subroutines, dealing with expansion or iteration methods for the generation of noninteger powers of a Chebyshev series, are somewhat more sophisticated than the first six subroutines, but the theoretical basis of their operation has recently been described in detail [1]. For this reason, they will not be described in more detail here. All four subroutines use the multiplication subroutine *MLTPLY* but are otherwise independent. The subroutines *BINOM*, *XALFA*2, and *XALFA*3 all have the same purpose but operate with different methods and have different convergence properties. All three are given in order to allow the user to experiment and eventually select the one that is most efficient for his particular application.

Acknowledgments. I wish to thank Nancy Hamata at the Jet Propulsion Laboratory for her assistance in the programming and debugging of the present subroutines; also the two anonymous reviewers for their useful suggestions.

References

1. Broucke, R. Construction of national and negative powers of a formal series. *Comm. ACM 14*, 1 (Jan. 1971), 32–35.
2. Carpenter, L. Planetary perturbations in Chebyshev series. NASA Tech. Note TN-D-3168, Goddard Space Flight Center, Greenbelt, Md., Jan. 1966.
3. Clenshaw, C. W. The numerical solution of linear differential equations in Chebyshev series. *Proc. Cambr. Phil. Soc., 53* (1957), 134–149.
4. Clenshaw, C.W., Chebyshev series for mathematical functions. *Nat. Phys. Lab. Math. Tables, 5* (1962) London, HMSO.
5. Clenshaw, C.W., and Norton, H.J. The solution of nonlinear ordinary differential equations in Chebyshev series. *Computer J. 6* (1963), 88–92.

Algorithm

```
      SUBRØUTINE CHEBY(NF, NPL, NPLMAX, N2, FUNCTN, X, FXJ, GC)
C SIMULTANEOUS CHEBYSHEV ANALYSIS ØF NF FUNCTIØNS
C CØMPUTES A MATRIX, X, CØNTAINING ØNE CHEBYSHEV SERIES PER
C CØLUMN FØR A GIVEN NUMBER ØF FUNCTIØNS, NF. INPUT NPL,
C THE NUMBER ØF TERMS IN ALL SERIES, NPLMAX, THE RØW
C DIMENSIØN ØF X IN THE CALLING PRØGRAM (MUST BE.GE.NPL),
C N2, DIMENSIØN ØF GC (MUST BE.GE.2*(NPL-1)), AND FUNCTN,
C THE NAME ØF USER SUBRØUTINE WHICH DEFINES THE NF
C FUNCTIØNS. FXJ AND GC ARE WØRK SPACE.
C AN EXAMPLE ØF SUCH A SUBRØUTINE IS AS FØLLØWS
C SUBRØUTINE FUNCTN(A,VAL)
C DØUBLE PRECISIØN A,VAL(2)
C VAL(1)=DSIN(A)
C VAL(2)=DCØS(A)
C RETURN
C END
      DØUBLE PRECISION X(NPLMAX,NF), FXJ(NF), GC(N2), ENN, XJ,
     * FK, PEN, FAC
      DØ 20 K=1,NPL
        DØ 10 J=1,NF
          X(K,J) = 0.D0
   10   CØNTINUE
   20 CØNTINUE
      N = NPL - 1
      ENN = N
      PEN = 3.141592653589793 2D0/ENN
      DØ 30 K=1,N2
        FK = K - 1
        GC(K) = DCØS(FK*PEN)
   30 CONTINUE
      DØ 80 J=1,NPL
        XJ = GC(J)
        CALL FUNCTN(XJ, FXJ)
        IF (J.NE.1 .AND. J.NE.NPL) GØ TØ 50
        DØ 40 K=1,NF
          FXJ(K) = .5D0*FXJ(K)
   40   CØNTINUE
   50   DØ 70 L=1,NPL
          LM = MØD((L-1)*(J-1),N2) + 1
          DØ 60 K=1,NF
            X(L,K) = X(L,K) + FXJ(K)*GC(LM)
   60     CØNTINUE
   70   CØNTINUE
   80 CØNTINUE
      FAC = 2.0D0/ENN
      DØ 100 K=1,NPL
        DØ 90 J=1,NF
          X(K,J) = FAC*X(K,J)
   90   CØNTINUE
  100 CØNTINUE
      RETURN
      END
```

```
      SUBRØUTINE MLTPLY(XX, X2, NPL, X3)
C MULTIPLIES TWØ GIVEN CHEBYSHEV SERIES, XX AND X2, WITH
C NPL TERMS TØ PRØDUCE AN ØUTPUT CHEBYSHEV SERIES, X3.
      DØUBLE PRECISIØN XX(NPL), X2(NPL), X3(NPL), EX
      DØ 10 K=1,NPL
        X3(K) = 0.0D0
   10 CØNTINUE
      DØ 30 K=1,NPL
        EX = 0.0D0
        MM = NPL - K + 1
        DØ 20 M=1,MM
          L = M + K - 1
          EX = EX + XX(M)*X2(L) + XX(L)*X2(M)
   20   CØNTINUE
        X3(K) = 0.5D0*EX
   30 CØNTINUE
      X3(1) = X3(1) - 0.5D0*XX(1)*X2(1)
      DØ 50 K=3,NPL
        EX = 0.0D0
        MM = K - 1
        DØ 40 M=2,MM
          L = K - M + 1
          EX = EX + XX(M)*X2(L)
   40   CØNTINUE
        X3(K) = 0.5D0*EX + X3(K)
   50 CØNTINUE
      RETURN
      END
```

```
      SUBRØUTINE ECHEB(X, CØEF, NPL, FX)
C EVALUATES THE VALUE FX(X) ØF A GIVEN CHEBYSHEV SERIES,
C CØEF, WITH NPL TERMS AT A GIVEN VALUE ØF X BETWEEN
C -1. AND 1.
      DØUBLE PRECISIØN CØEF(NPL), X, FX, BR, BRPP, BRP2
      BR = 0.0D0
      BRPP = 0.0D0
      DØ 10 K=1,NPL
        J = NPL - K + 1
        BRP2 = BRPP
        BRPP = BR
        BR = 2.0D0*X*BRPP - BRP2 + CØEF(J)
   10 CØNTINUE
      FX = 0.5D0*(BR-BRP2)
      RETURN
      END
```

```
      SUBRØUTINE EDCHEB(X, CØEF, NPL, FX)
C EVALUATES THE VALUE FX(X) ØF THE DERIVATIVE ØF A
C CHEBYSHEV SERIES, CØEF, WITH NPL TERMS AT A GIVEN
C VALUE ØF X BETWEEN -1. AND 1.
      DØUBLE PRECISIØN CØEF(NPL), X, FX, XJP2, XJPL, XJ, BJP2,
     * BJPL, BJ, BF, DJ
      XJP2 = 0.0D0
      XJPL = 0.0D0
      BJP2 = 0.0D0
      BJPL = 0.0D0
      N = NPL - 1
      DØ 10 K=1,N
        J = NPL - K
        DJ = J
        XJ = 2.D0*CØEF(J+1)*DJ + XJP2
        BJ = 2.D0*X*BJPL - BJP2 + XJ
        BF = BJP2
        BJP2 = BJPL
        BJPL = BJ
        XJP2 = XJPL
        XJPL = XJ
   10 CØNTINUE
      FX = .5D0*(BJ-BF)
      RETURN
      END
```

```
      SUBRØUTINE DFRNT(XX, NPL, X2)
C CØMPUTES THE DERIVATIVE CHEBYSHEV SERIES, X2, ØF A GIVEN
C CHEBYSHEV SERIES, XX, WITH NPL TERMS.
C TØ REPLACE A SERIES X BY ITS DERIVATIVE, USE
C CALL DFRNT(X,NPL,X)
      DØUBLE PRECISIØN XX(NPL), XXN, XXL, DN, DL, X2(NPL)
      DN = NPL - 1
      XXN = XX(NPL-1)
      X2(NPL-1) = 2.D0*XX(NPL)*DN
      X2(NPL) = 0.D0
      DØ 10 K=3,NPL
        L = NPL - K + 1
        DL = L
        XXL = XX(L)
        X2(L) = X2(L+2) + 2.D0*XXN*DL
        XXN = XXL
   10 CØNTINUE
      RETURN
      END


      SUBRØUTINE NTGRT(XX, NPL, X2)
C CØMPUTES THE INTEGRAL CHEBYSHEV SERIES, X2, ØF A GIVEN
C CHEBYSHEV SERIES, XX, WITH NPL TERMS.
C TØ REPLACE A SERIES X BY ITS INTEGRAL, USE
C CALL NTGRT(X,NPL,X)
      DØUBLE PRECISIØN XX(NPL), XPR, TERM, DK, X2(NPL)
      XPR = XX(1)
      X2(1) = 0.0D0
      N = NPL - 1
      DØ 10 K=2,N
        DK = K - 1
        TERM = (XPR-XX(K+1))/(2.D0*DK)
        XPR = XX(K)
        X2(K) = TERM
   10 CØNTINUE
      DK = N
      X2(NPL) = XPR/(2.D0*DK)
      RETURN
      END


      SUBRØUTINE INVERT(X, XX, NPL, NET, XNVSE, WW, W2)
C CØMPUTES THE INVERSE CHEBYSHEV SERIES, XNVSE, GIVEN A
C CHEBYSHEV SERIES, X, A FIRST APPRØXIMATIØN CHEBYSHEV
C SERIES, XX, WITH NPL TERMS, AND THE NUMBER ØF
C ITERATIØNS, NET. THE SUBRØUTINE USES THE EULER METHØD
C AND CØMPUTES ALL PØWERS EPS**K UP TØ K=2**(NET+1),
C WHERE EPS=1-X*(XX INVERSE). WW AND W2 ARE WØRK SPACE.
C SUBRØUTINES USED - MLTPLY
      DØUBLE PRECISIØN X(NPL), XX(NPL), XNVSE(NPL), WW(NPL),
     * W2(NPL)
      CALL MLTPLY(X, XX, NPL, WW)
      WW(1) = 2.D0 - WW(1)
      DØ 10 K=2,NPL
        WW(K) = -WW(K)
   10 CØNTINUE
      CALL MLTPLY(WW, WW, NPL, W2)
      WW(1) = 2.D0 + WW(1)
      DØ 40 K=1,NET
        CALL MLTPLY(WW, W2, NPL, XNVSE)
        DØ 20 J=1,NPL
          WW(J) = WW(J) + XNVSE(J)
   20   CØNTINUE
        CALL MLTPLY(W2, W2, NPL, XNVSE)
        DØ 30 J=1,NPL
          W2(J) = XNVSE(J)
   30   CØNTINUE
   40 CØNTINUE
      CALL MLTPLY(WW, XX, NPL, XNVSE)
      RETURN
      END


      SUBRØUTINE BINØM(X, XX, NPL, M, NT, XA, WW, W2, W3)
C CØMPUTES THE BINØMIAL EXPANSIØN SERIES, XA, FØR (-1/M)
C PØWER ØF A GIVEN CHEBYSHEV SERIES, X, WITH NPL TERMS,
C WHERE M IS A PØSITIVE INTEGER. XX IS A GIVEN INITIAL
C APPRØXIMATIØN TØ X**(-1/M). NT IS A GIVEN NUMBER ØF
C TERMS IN BINØMIAL SERIES. WW, W2, AND W3 ARE WØRK SPACE
C SUBRØUTINES USED - MLTPLY
      DØUBLE PRECISIØN X(NPL), XX(NPL), XA(NPL), WW(NPL),
     * W2(NPL), W3(NPL), ALFA, CØEF, DM, DKMM, DKM2
      DM = M
      ALFA = -1.D0/DM
      DØ 10 J=1,NPL
        WW(J) = X(J)
   10 CØNTINUE
      DØ 30 K=1,M
        CALL MLTPLY(WW, XX, NPL, W2)
        DØ 20 J=1,NPL
          WW(J) = W2(J)
   20   CØNTINUE
   30 CØNTINUE
      WW(1) = WW(1) - 2.D0
      XA(1) = 2.D0
      DØ 40 J=2,NPL
        XA(J) = 0.0D0
        W3(J) = 0.D0
   40 CØNTINUE
      W3(1) = 2.D0
      DØ 60 K=2,NT
        DKMM = K - 1
        DKM2 = K - 2
        CØEF = (ALFA-DKM2)/DKMM
        CALL MLTPLY(W3, WW, NPL, W2)
        DØ 50 J=1,NPL
          W3(J) = W2(J)*CØEF
          XA(J) = XA(J) + W3(J)
   50   CØNTINUE
   60 CØNTINUE
      CALL MLTPLY(XA, XX, NPL, W2)
      DØ 70 J=1,NPL
        XA(J) = W2(J)
   70 CØNTINUE
      RETURN
      END


      SUBRØUTINE XALFA2(X, XX, NPL, M, MAXET, EPSLN, NET, WW,
     * W2)
C REPLACES A GIVEN INITIAL APPRØXIMATIØN CHEBYSHEV SERIES,
C XX, BY A GIVEN CHEBYSHEV SERIES, X, WITH NPL TERMS,
C RAISED TØ THE (-1/M) PØWER, WHERE M IS AN INTEGER.
C INPUT MAXET, MAXIMUM ALLØWED NUMBER ØF ITERATIØNS, AND
C EPSLN, REQUIRED PRECISIØN EPSILØN. ØUTPUT ARGUMENT,
C NET, IS NUMBER ØF ITERATIØNS PREFØRMED. IF MAXET=NET,
C REQUIRED PRECISIØN MAY NØT HAVE BEEN REACHED AND THERE
C MAY BE DIVERGENCE. WW AND W2 ARE WØRK SPACE.
C CØNVERGENCE IS QUADRATIC
C SUBRØUTINES USED - MLTPLY
      DØUBLE PRECISIØN X(NPL), XX(NPL), WW(NPL), W2(NPL),
     * EPSLN, DALFA, DM, S, TDMM
      DM = M
      DALFA = 1.D0/DM
      TDMM = 2.D0*(DM+1.D0)
      DØ 60 JX=1,MAXET
        DØ 10 L=1,NPL
          WW(L) = X(L)
   10   CØNTINUE
        DØ 30 K=1,M
          CALL MLTPLY(WW, XX, NPL, W2)
          DØ 20 L=1,NPL
            WW(L) = W2(L)
   20     CØNTINUE
   30   CØNTINUE
        S = -2.D0
        DØ 40 L=1,NPL
          S = S + DABS(WW(L))
          WW(L) = -WW(L)
   40   CØNTINUE
        WW(1) = WW(1) + TDMM
        CALL MLTPLY(WW, XX, NPL, W2)
        DØ 50 L=1,NPL
          XX(L) = W2(L)*DALFA
   50   CØNTINUE
        NET = JX
        IF (DABS(S).LT.EPSLN) RETURN
   60 CØNTINUE
      RETURN
      END


      SUBRØUTINE XALFA3(X, XX, NPL, M, MAXET, EPSLN, NET, WW,
     * W2)
C REPLACES A GIVEN INITIAL APPRØXIMATIØN CHEBYSHEV SERIES,
C XX, BY A GIVEN CHEBYSHEV SERIES, X, WITH NPL TERMS,
C RAISED TØ THE (-1/M) PØWER, WHERE M IS AN INTEGER.
C INPUT MAXET, MAXIMUM ALLØWED NUMBER ØF ITERATIØNS, AND
C EPSLN, REQUIRED PRECISIØN EPSILØN. ØUTPUT ARGUMENT,
C NET, IS NUMBER ØF ITERATIØNS PREFØRMED. IF MAXET=NET,
C REQUIRED PRECISIØN MAY NØT HAVE BEEN REACHED AND THERE
C MAY BE DIVERGENCE. WW AND W2 ARE WØRK SPACE.
C CØNVERGENCE IS ØF ØRDER THREE
C SUBRØUTINES USED - MLTPLY
      DØUBLE PRECISIØN X(NPL), XX(NPL), WW(NPL), W2(NPL),
     * EPSLN, DALFA, DM, S, TDMM, P5DML
      DM = M
      DALFA = 1.D0/DM
      TDMM = 2.D0*(DM+1.D0)
      P5DML = .5D0*(DM+1.D0)
      DØ 90 JX=1,MAXET
        DØ 10 L=1,NPL
          WW(L) = X(L)
   10   CØNTINUE
        DØ 30 K=1,M
          CALL MLTPLY(WW, XX, NPL, W2)
          DØ 20 L=1,NPL
            WW(L) = W2(L)
   20     CØNTINUE
   30   CØNTINUE
        S = -2.D0
        DØ 40 L=1,NPL
          S = S + DABS(WW(L))
   40   CØNTINUE
        WW(1) = WW(1) - 2.D0
        DØ 50 L=1,NPL
          WW(L) = WW(L)*DALFA
   50   CØNTINUE
        CALL MLTPLY(WW, WW, NPL, W2)
        DØ 60 L=1,NPL
          WW(L) = -WW(L)
          W2(L) = W2(L)*P5DML
   60   CØNTINUE
        WW(1) = WW(1) + 2.D0
        DØ 70 L=1,NPL
          W2(L) = W2(L) + WW(L)
   70   CØNTINUE
        CALL MLTPLY(W2, XX, NPL, WW)
        DØ 80 L=1,NPL
          XX(L) = WW(L)
   80   CØNTINUE
        NET = JX
        IF (DABS(S).LT.EPSLN) RETURN
   90 CØNTINUE
      RETURN
      END
```