NATIONAL RESEARCH COUNCIL OF CANADA
HERZBERG INSTITUTE OF ASTROPHYSICS

PUBLICATIONS

of the

DOMINION ASTROPHYSICAL OBSERVATORY

VICTORIA, B.C.
Volume XVI, No. 6

INTEP, AN EFFECTIVE INTERPOLATION SUBROUTINE

by

GRAHAM HILL

ABSTRACT: The computer code for a most useful interpolation subroutine is presented, based on a little-known algorithm by Tsipouras and Cormier (1973).

RÉSUMÉ: Le code informatique d'un sous-programme d'interpolation des plus utiles, fondé sur un algorithme peu connu de Tsipouras et Cormier (1973), est presénté.

I. INTRODUCTION

In this paper I describe a computer program (INTEP) which is a remarkably effective interpolation routine. Because it has proved to be extremely useful I have been encouraged to publish the computer code, thereby making it available for widespread use. The mathematical source of the program is a little-known paper called "Hermite interpolation algorithm for constructing reasonable analytic curves through discrete data points" by Tsipouras and Cormier (1973). I present this program to draw the attention of the astronomical community to this algorithm in the hope that they will forsake the use of polynomials when it is desired to draw "reasonable curves" through tabular or experimental data points.

II. THE PROGRAM

General Comments

INTEP is based on a modification of the Hermite interpolation formula (see Hildebrand 1974, page 377; Johnson and Riess 1977, page 196) by Tsipouras and Cormier (1973) who used it to draw what they describe as "reasonable curves" through a variety of meteorological data. The main constraint of their algorithm is the requirement that the derived interpolated curve should match what one might subjectively draw by hand through the same data points. The examples they use amply illustrate the power of their algorithm over Lagrangian interpolation polynomials (see Hildebrand pages 80, 159) and least-squares fits of polynomials to data. The least-squares fitting of polynomials to data is the most common interpolation procedure used by astronomers. Such fits, however, most often fail at extrema. This is a well-known problem with polynomials (see Tsipouras and Cormier; Johnson and Riess 1977, page 201) and it causes difficulty when, for example, one is attempting to rectify spectra. In this case one wants the fitted curve to go through certain continuum points but the user usually settles for minimal excursions about these points since polynomial interpolation will rarely fit this type of data. Two examples, typical of the kind of non-analytical functions which INTEP handles so well, are shown in Figures 1 and 2. Here reasonable curves are drawn through a H-D (characteristic) curve (log Intensity vs log Density) in Figure 1 and through a simulated spectral continuum in Figure 2. For comparison, high-order polynomial fits are also shown. In these examples, contrary to those of Tsipouras and Cormier, the differences between the alternative schemes are small but they adequately show the difficulties.

Program Outline

The FORTRAN code is given in Table 1 with detailed instructions as to its use. The data may be in ascending or descending order. For random interpolation calls one should simply call INTEP. When rectifying a spectrum for example the calculations can be expedited by calling INTEP initially and thereafter entering the subroutine through EINTEP. As an example of the speed gained in this way INTEP was run on an array of 16 elements with 10^5 interpolation steps. With no ENTRY calls the execution time on a VAX 11/780 was 32 seconds, but with the ENTRY call the time was 13 seconds.

I wish to thank Tony Lynas-Gray for his enthusiastic response to INTEP and for insisting that the code be published. Steve Morris had many useful comments on the manuscript; I thank him.

Dominion Astrophysical Observatory,
Victoria, B.C.
September, 1982.

REFERENCES

Hildebrand, F.B. 1974, Introduction to Numerical Analysis (2nd ed., McGraw-Hill: New York)
Johnson, L.W., and Riess, R.D. 1977, Numerical Analysis (Addison-Wesley Publ. Co.: Mass)
Tsipouras, J., and Cormier, R.V. 1973, Air Force Surveys in Geophysics, No. 272.

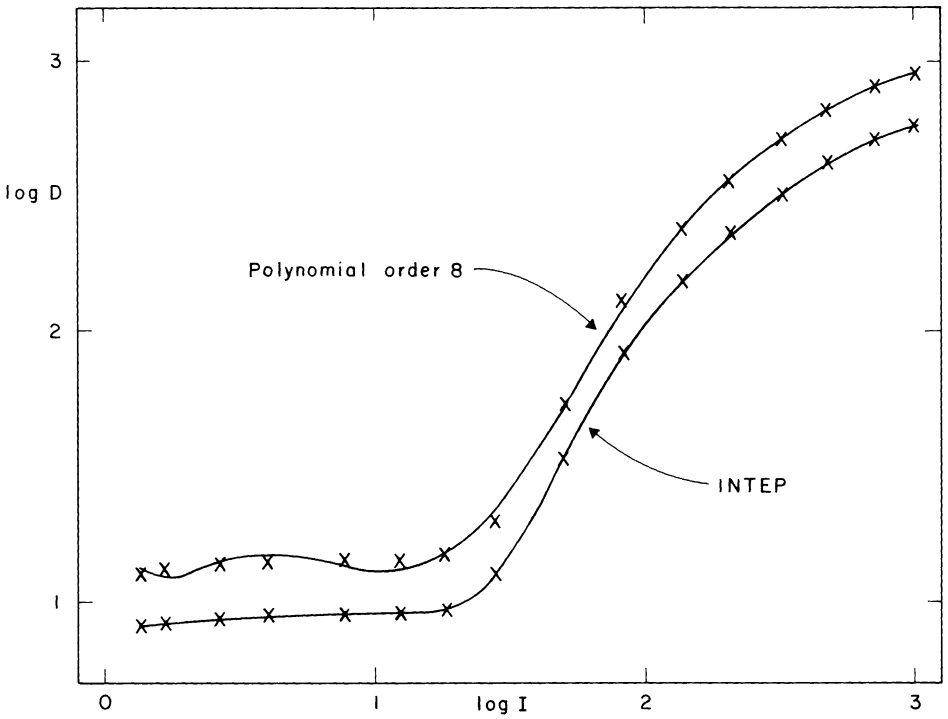

Figure 1. Comparison between INTEP and a high order polynomial with reference to a characteristic curve.

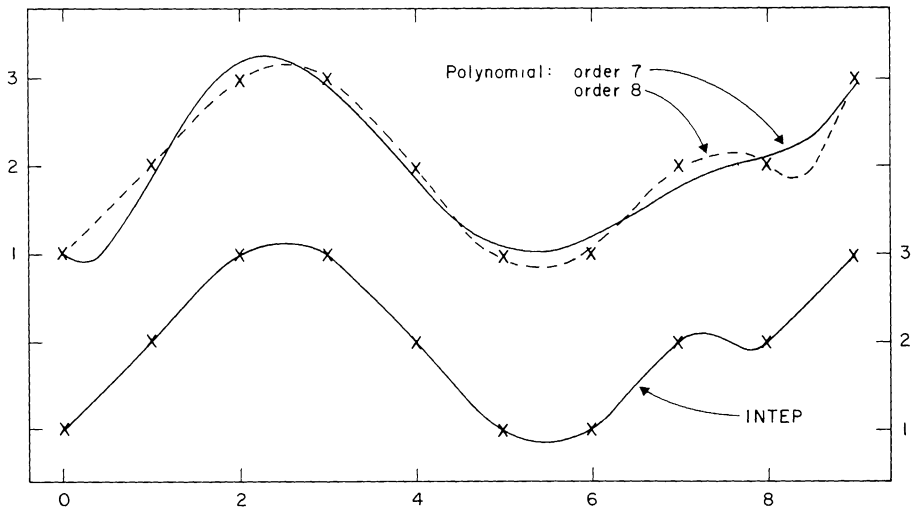

Figure 2. Comparison between INTEP and two high order polynomials with reference to simulated continuum data.

TABLE 1. INTEP

```
      SUBROUTINE INTEP(XP,P,X,F,N,IER)
C     Purpose:
C     To interpolate a function value P for a given argument
C     value XP using a table of N values (X,F).
C     This is a Spline Interpolation scheme based on Hermite polynomials.
C     The source is U.S. Airforce Surveys in Geophysics No 272.
C     Usage:
C     For random values of XP
C      CALL INTEP(XP,P,X,F,N,IER)
C     or after the first call to INTEP with monotonically increasing
C     or decreasing values of XP consistent with the X vector
C      CALL EINTEP(XP,P,X,F,N,IER)
C     Description of parameters:
C     XP    The chosen argument value.
C     P     The resultant interpolated value.
C     X     The vector of independent values.
C     F     The vector of function or dependent values.
C     N     The number of points in the (X,P) vectors.
C     IER   The resultant error parameter.
C     Remarks:
C     If XP is beyond either extreme in the vector X the value of F
C     at that extreme is adopted and IER set to 2.
      REAL LP1,LP2,L1,L2
      DIMENSION F(1),X(1)
      IER=1
      IO=1
      IUP=0
      IF(X(2).LT.X(1))IUP=1
      N1=N-1
      IF((XP.GE.X(N).AND.IUP.EQ.0).OR.(XP.LE.X(N).AND.IUP.EQ.1))THEN
    5  P=F(N)
       GO TO 6
      ELSE IF((XP.LE.X(1).AND.IUP.EQ.0).OR.
     * (XP.GE.X(1).AND.IUP.EQ.1))THEN
       P=F(1)
    6  IER=2
       RETURN
      END IF
      ENTRY EINTEP(XP,P,X,F,N,IER)
    8 DO 1 I=IO,N
       IF(XP.LT.X(I).AND.IUP.EQ.0)GO TO 2
       IF(XP.GT.X(I).AND.IUP.EQ.1)GO TO 2
    1 CONTINUE
      GO TO 5
    2 I=I-1
      IF(I.EQ.IO-1)GO TO 4
      IO=I+1
      LP1=1./(X(I)-X(I+1))
      LP2=1./(X(I+1)-X(I))
      IF(I.EQ.1)FP1=(F(2)-F(1))/(X(2)-X(1))
      IF(I.EQ.1)GO TO 3
      FP1=(F(I+1)-F(I-1))/(X(I+1)-X(I-1))
    3 IF(I.GE.N1)FP2=(F(N)-F(N-1))/(X(N)-X(N-1))
      IF(I.GE.N1)GO TO 4
      FP2=(F(I+2)-F(I))/(X(I+2)-X(I))
    4 XPI1=XP-X(I+1)
      XPI=XP-X(I)
      L1=XPI1*LP1
      L2=XPI*LP2
      P=F(I)*(1.-2.*LP1*XPI)*L1*L1+F(I+1)*(1.- 2.*LP2*XPI1)
     1*L2*L2+FP2*XPI1*L2*L2 +FP1*XPI*L1*L1
      RETURN
      END
```