12.1 One-Dimensional Table Look Up, Interpolation, and Differentiation

A. Purpose

Given a table, $(x_i,\ y_i)$, of independent variable values and the corresponding dependent variable values, this subroutine finds the points in the table closest to a given value of the independent variable and uses these points to interpolate for the corresponding value of the dependent variable. Error estimates, Hermite interpolation (i.e. using $(x_i,\ y_i,\ y_i'))$, different look up methods, and the computation of derivative information are available.

B. Usage

B.1 Program Prototype, Single Precision

INTEGER NTAB, NDEG, LUP, IOPT$(\geq k_1)$
[$k_1 \geq 2$ depends on options used.]

REAL X, Y, XT($\geq$NTAB)**, YT**($\geq$NTAB)**,**
EOPT$(\geq k_2)$ [$k_2 \geq 1$ depends on options used .]

CALL SILUP (X, Y, NTAB, XT, YT, NDEG, LUP, IOPT, EOPT)

B.2 Argument Definitions

X [in] Independent variable where value of interpolant is desired.

Y [out] Value of interpolant.

NTAB [in] Number of points in the table, $\geq$ NDEG+k, where $k = 1$ if NDEG is odd and 2 if NDEG is even, and must be one bigger than this if an error estimate is requested.

XT() [in] Array of independent variable values, XT(i) = x_i. Must be monotone increasing or monotone decreasing, but this is not checked for. (XT(I) = XT(I+1) is permitted, but has a special meaning, see Section C.) If the x_i are equally spaced one should set LUP (below) to 3, and provide x_1 in XT(1) and $x_2 - x_1$ in XT(2).

YT() [in] Array of dependent variable values. YT(i) = y_i.

NDEG [in] Nominal degree of the polynomial to be used in the interpolation. Require $0 \leq$ NDEG ≤ 15.

For NDEG = 0, Y is set to YT(k), where k minimizes $|$X $- x_k|$.

For NDEG odd, Y is the result of standard polynomial interpolation of degree NDEG using NDEG + 1 points. The interpolating function is continuous, but the first derivative is usually discontinuous at tabular points.

For NDEG > 0 and even, Y is obtained from a function that is a linear combination of two polynomials of degree NDEG. This function is a polynomial of degree NDEG + 1 using NDEG + 2 points that interpolates all but the two outer points. It has a continuous first derivative but the second derivative is usually discontinuous at tabular points.

Usually $|error|$ in the interpolant will tend to decrease in a fairly regular way for increasing values of NDEG until either errors due to a lack of precision in YT, or the inherent instability of high degree polynomial interpolation causes errors to get worse. If one wants a continuous first derivative, even values of NDEG > 0 should be used.

LUP [inout] Defines the type of look up method. (Changed only if LUP ≤ 0 on input.)

- ≤ 0 If LUP = 0, start with a binary search, else do a sequential search starting with an index of $-$LUP. On exit, LUP is set to $-k$, where k minimizes $|$X $-$ XT(k)$|$.
- $= 1$ Binary search. Use this when accesses are not sequential, and XT values are not close to being equally spaced.
- $= 2$ Start a sequential search with an index = [1.5 + (NTAB$-$1) (X $-$ XT(1))/(XT(NTAB) $-$ XT(1))]. Use this when points are almost equally spaced and accesses are not sequential.
- $= 3$ YT(I) corresponds to XT(1) + (I$-$1)*XT(2), no search is required to do the look up and XT can have dimension XT(2). This is recommended when the XT values are equally spaced since it takes the least space and should usually be fastest.
- $= 4$ Internal information connected with the x_k values used in the last interpolation is reused. (Don't use this value if there is any chance for an intervening call to this subroutine.) Only YT should be changed; IOPT is not examined. This option saves time when interpolating components after the first of a vector valued function.

IOPT() [inout] IOPT(1) is used to return a status as follows:

-10 An option index is out of range.

−9 NTAB is outside allowed limits.

−8 NDEG is outside allowed limits.

−7 LUP > 3 when program was not ready for it.

−6 Option 3 (compute derivatives), has requested more than 15 derivatives.

−5 LUP = 3, and XT(2) = 0.

−4 XT(1) = XT(NTAB), and NTAB is not 1.

−3 Bad points (see option 6) mean only 0 or 1 points were available for interpolation..

−2 There is only one table entry; the estimated error that was requested was not computed.

−1 The accuracy requested was not obtained.

0 Normal return, no exceptional conditions.

1 X was outside the domain of the table, extrapolation used.

2 Available table values were so few that this restricted the degree of the polynomial. A valid error estimate is not returned in this case.

Starting with IOPT(2) options are specified by integers in the range 0 to 7, followed in some cases by integers providing argument(s) for the option. Each option, with its arguments if any, is followed in IOPT by the next option or by a 0. If an option index is specified more than once, only the last specification is used.

0 End of the option list; this must always be the last option specified in IOPT.

1 An error estimate is to be returned in EOPT(1).

2 (Argument: K2) K2 gives the polynomial degree to use when extrapolating. The default for K2 is NDEG if NDEG $\leq$ 2 or if NDEG is even, else it is NDEG − 1.

3 (Arguments: K3, L3) Save (k^{th} derivative of interpolating polynomial)/k! in EOPT(K3+k−1) for k = 1, 2, ..., L3. These values are the coefficients of the polynomial in the monomial basis expanded about X. Require $0 \leq$ L3 ≤ 15.

4 (Argument K4) The absolute and relative errors expected in YT entries are specified in EOPT(K4) and EOPT(K4+1) respectively. The values provided here are used in estimating the error in the interpolation. An error estimate is returned in EOPT(1).

5 (Argument K5, L5) Do the interpolation to the accuracy requested by the absolute error tolerance specified in EOPT(K5) and the relative error tolerance in EOPT(K5+1) respectively. An attempt is made to keep the final error < EOPT(K5) + EOPT(K5+1) (|YT(k)| + |YT(k')|), where k and k' are indices for table values close to X. Standard polynomial interpolation is done, but here NDEG gives the maximal degree polynomial to use in the interpolation. The actual degree used in doing the interpolation is stored in the space for the argument L5. If both EOPT's specified are ≤ 0, IOPT(1) is not set to −1, and no error message is generated due to an unsatisfied accuracy request. An error estimate is returned in EOPT(1).

6 (Argument K6) Do not use point (XT(k), YT(k)) in the interpolation if YT(k) = EOPT(K6). This option is useful if one has a table with equally spaced points, but with some bad data points, and may also be used by the multiple dimensional interpolation subroutine. Points are selected for use in the interpolation as if points flagged with YT(k) = EOPT(K6) were not present.

7 (Argument K7) YT(K7+k) gives the first derivative corresponding to the function value in YT(k). These derivatives are to be used in doing the interpolation. One gets a continuous interpolant only for NDEG = 3, 7, 11, and 15; these cases also give a continuous first derivative in the interpolant. The interpolating polynomial satisfies p(XT(k)) = YT(k), p'(XT(k)) = YT(K7+k) for values of k that give values of XT close to X. (Section D describes how points are selected.) If NDEG is even, a value of YT(k) is used without using the corresponding value of YT(K7+k).

EOPT [inout] Array used to return an error estimate and used for options.

EOPT(1) [out] contains an estimate of the error in the interpolation if an error estimate has been requested by setting option 1, 4 or 5.

EOPT(> 1) [in or out] for use by options 3–6.

B.3 Modifications for Double Precision

Change SILUP to DILUP, and the REAL type statement to DOUBLE PRECISION.

C. Examples and Remarks

DRSILUP below is a sample program that interpolates in a table of $\sin(x)$ given with a spacing of 0.5 for NDEG = 2 to 10, and obtains an error estimate. ODSILUP below gives the results of running DRSILUP on an IBM PC, which uses IEEE 32 bit floating point arithmetic.

The user is reminded that polynomial interpolation of high degree is hazardous, where "high" depends strongly on the kind of data being interpolated. The error estimates provided by the program are usually greater than the actual error, but on any given interpolation may be much too small. Similarly, the order selected by the program to satisfy a given error criterion will usually do a good job, but will (infrequently) use too low a degree due to an overly optimistic error estimate or an overly pessimistic conclusion that the corrections are starting to diverge. When many interpolations are going to be done in a table we recommend trying several degrees at enough points to cover the kinds of functional behavior in different areas of the table and examining the error estimates. Then later interpolations can be done using the fixed degree that appears best for the accuracy desired.

If one has a table with discontinuities, one can set XT(I) = XT(I+1) = the point of the discontinuity. The x_j's that are used in the interpolation will be selected so that all j satisfy $j \leq$ I, or $j \geq$ I + 1. If the discontinuity lies between two points, use option 6 to define the value at the point of discontinuity.

D. Functional Description

The look up process identifies the point in the table, XT(k), nearest to the input value X. To obtain continuity in the interpolant, remaining points are selected one at a time, keeping the number of points on either side of X balanced as long as this is possible. Thus XT(k) and XT(k'), the second entry selected, will bracket X if extrapolation is not required. When NDEG > 0 is even (and Hermite interpolation is not being used), a linear combination of the polynomials of degree NDEG interpolating the left NDEG + 1 and the right NDEG + 1 points is used. (A total of NDEG + 2 points is used.) If it is not possible to select (NDEG + 2)/2 points on either side of X, then just NDEG + 1 points are used to obtain a polynomial of degree NDEG as in the odd case. When extrapolating, the default action is to use standard polynomial interpolation of degree $2 \times \max(1, \lfloor \text{NDEG}/2 \rfloor)$. This default can be changed using an option value of 2 in IOPT(). Interpolations are done using the Newton divided difference form of the interpolating polynomial as described for Algorithms I (NDEG odd) and IV (NDEG> 0, even) by Krogh in [1]. The derivatives are computed as described for Algorithm V in [1].

Define E_{Min} by the following:

$$E_{Min} = E_{Abs} + E_{Rel}\,(|\text{YT}(i)| + |\text{YT}(i')|),$$

where E_{Abs} = EOPT(K4) if option 4 is used, and is otherwise 0; E_{Rel} = EOPT(K4+1) if option 4 is used and is otherwise the smallest positive floating point number that gives a number different from 1 when added to 1; and i and i' are the indices for the first two table entries selected for use in the interpolation.

Then the error is estimated by

$$\text{error est.} = 1.5\left(|y_c - P_{k+1}| + \frac{|P_k - P_{k-1}|}{32}\right) + E_{Min},$$

where y_c is the value being returned for Y, P_k is the result obtained by standard polynomial interpolation of degree k, and k is NDEG unless the degree has been reduced because of extrapolation in which case k is the degree actually used. If there are not sufficient points available to compute P_{k+1}, then the error estimate is given by $1.5\ |P_k - P_{k-1}| + E_{Min}$.

The choice of degree for option 5 makes use of the following quantities.

$$e_0 = |P_0|$$
$$\overline{e}_1 = \max(.75,\ e_1/(e_1 + e_0)$$
$$e_j = |P_j - P_{j-1}|, \quad j = 1,\ 2,\ \dots$$
$$\overline{e}_j = \frac{1}{2}\overline{e}_{j-1} + e_j/(e_j + e_{j-1}), \quad j = 2, 3,\ \dots$$
$$\widehat{e}_j = \frac{1}{2}e_{j-1}\overline{e}_{j-1}/j, \quad j = 2,\ 3,\ \dots$$

The value of j is allowed to increase until either $e_j + e_{j-1} \leq$ error request, or $e_j \geq \widehat{e}_j$ for two successive steps with $j > 1$. The quantity, $\overline{e}_j$ indicates the rate of decrease in e_j for successive values of j. A good rate of decrease in the past causes a better rate to be required in the future. Also, better convergence is required as the degree gets larger. The final error estimate is given by $1.5\ (e_j + .0625e_{j-1}) + E_{Min.}$ When derivatives are being computed, the error request is decreased by the factor $|(x - x_i)/(x_i - x_{i'})|$, where i and i' are the indices for the first two table entries selected.

References

1. Fred T. Krogh, *Efficient algorithms for polynomial interpolation and numerical differentiation*, **Math. of Comp. 24**, 109 (Jan. 1970) 185–190.

E. Error Procedures and Restrictions

Values of IOPT(1) < 0 ordinarily cause an error message to be printed, and those < -3 do not ordinarily result in a return to the user. One can change the action on errors by calling the message/error routine MESS of Chapter 19.3 before calling this routine.

F. Supporting Information

The source language is ANSI Fortran 77.

Entry	**Required Files**
DILUP	AMACH, DILUP, DMESS, MESS
SILUP	AMACH, MESS, SILUP, SMESS

Subroutine designed and written by: Fred T. Krogh, JPL, May 1991. Hermite interpolation added December 1994.

DRSILUP

```
      program drsilu
c>> 2001-05-22 DRSILU Krogh Minor change for making .f90 version.
c>> 1994-12-21 DRSILU  Krogh  Latest version.
c--S replaces "?": DR?ILU, ?ILUP
c Demonstration driver for SILUP.
c Given table of sin(x), x = 0, .5, 1., ..., interpolates for x =
c -.1, .1, and, 8.3 using polynomial degrees 2, 3, 4, ..., 10.  Errors
c are estimated in all cases.
c
      integer NTAB, NX, NDEG1, NDEG2
      parameter (NTAB=40, NX=3, NDEG1=2, NDEG2=10)
      real            XT(2), YT(NTAB), X(NX), Y, EOPT(1), H, ANS
      parameter (H = .5E0)
      integer I, NDEG, IOPT(3), LUP
      data X / -.1E0, .1E0, 8.3E0 /
c                          Set IOPT to get an error estimate.
      data IOPT / 0, 1, 0 /
      data LUP / 3 /
c
c Compute the XT and YT tables
      XT(1) = 0.E0
      XT(2) = H
      do 10 I = 1, NTAB
         YT(I) = sin(real(I-1) * H)
   10 continue
      print *, 'IOP(1) NDEG     X           Y      Est. Error True Error'
      do 30 I = 1, NX
         ANS = sin(X(I))
         do 20 NDEG = NDEG1, NDEG2
            call SILUP (X(I), Y, NTAB, XT, YT, NDEG, LUP, IOPT, EOPT)
            print '(I5, I5, F9.4, F12.8, 1P,E10.2, E11.2)',
     1        IOPT(1), NDEG, X(I), Y, EOPT(1), Y - ANS
   20    continue
      print *
   30 continue
      stop
      end
```

ODSILUP

IOP(1)	NDEG	X	Y	Est. Error	True Error
1	2	−0.1000	−0.10997072	1.24E−02	−1.01E−02
1	3	−0.1000	−0.10997072	1.24E−02	−1.01E−02
1	4	−0.1000	−0.09861922	9.96E−04	1.21E−03
1	5	−0.1000	−0.09861922	9.96E−04	1.21E−03
1	6	−0.1000	−0.09992263	1.23E−04	−8.92E−05
1	7	−0.1000	−0.09992263	1.23E−04	−8.92E−05
1	8	−0.1000	−0.09984791	6.84E−05	−1.45E−05
1	9	−0.1000	−0.09984791	6.84E−05	−1.45E−05
1	10	−0.1000	−0.09982507	2.23E−05	8.35E−06
0	2	0.1000	0.10527554	6.82E−03	5.44E−03
0	3	0.1000	0.10102075	2.74E−03	1.19E−03
0	4	0.1000	0.09932593	4.38E−04	−5.07E−04
0	5	0.1000	0.09956475	4.60E−04	−2.69E−04
0	6	0.1000	0.09986382	4.69E−05	3.04E−05
0	7	0.1000	0.09988573	7.15E−05	5.23E−05
0	8	0.1000	0.09983876	2.29E−05	5.33E−06
0	9	0.1000	0.09982495	9.73E−06	−8.47E−06
0	10	0.1000	0.09983100	6.90E−06	−2.42E−06
0	2	8.3000	0.90053833	5.39E−04	−1.63E−03
0	3	8.3000	0.90091217	1.73E−03	−1.26E−03
0	4	8.3000	0.90208685	8.11E−05	−8.49E−05
0	5	8.3000	0.90210736	8.78E−05	−6.44E−05
0	6	8.3000	0.90216720	9.84E−07	−4.59E−06
0	7	8.3000	0.90216827	4.78E−06	−3.52E−06
0	8	8.3000	0.90217149	3.35E−07	−2.98E−07
0	9	8.3000	0.90217155	3.89E−07	−2.38E−07
0	10	8.3000	0.90217173	1.05E−07	−5.96E−08